DIMENSION REDUCTION FOR
HETEROGENEOUS POPULATIONS OF
OSCILLATORS

Tom S. Bertalan

A DISSERTATION
PRESENTED TO THE FACULTY
OF PRINCETON UNIVERSITY
IN CANDIDACY FOR THE DEGREE
OF DOCTOR OF PHILOSOPHY

RECOMMENDED FOR ACCEPTANCE
BY THE DEPARTMENT OF
CHEMICAL AND BIOLOGICAL ENGINEERING
ADVISER: PROFESSOR IOANNIS G. KEVREKIDIS

APRIL 2018
Abstract

This dissertation discusses coarse-graining methods and applications for simulations of large heterogeneous populations of neurons. These simulations are structured as large coupled sets of ordinary differential equations describing the state evolution for many qualitatively similar, but quantitatively distinct individual units. In full generality, the direct coupling between these units is not all-to-all, but is mediated through a directed network. With sufficiently strong coupling and weak heterogeneity across the population, a common outcome for such simulations is synchronization. Here, the states of individual units, while not identical, can be neatly approximated by a smooth function of some latent independent variable. It is this smooth structure that we seek to exploit in this dissertation for both didactic and computational purposes. Briefly put, the polynomial chaos expansion (PCE) methods used in this dissertation are reminiscent of Fourier expansions, recast in a setting where the spatial domain is a parameter space rather than a physical space, and, as such, has an associated probability density.

After describing methods and introducing a common terminology in chapter 1, we examine several applications in the remaining chapters, each illustrating different computational benefits and concerns for PCE. The applications range in complexity from the simple Kuramoto coupled phase oscillator model in chapter 3 to a biophysically realistic simulation of circadian rhythms with up to 23 dynamic quantities per unit. For each, some coarse computational technique(s) are demonstrated, including the computation of fixed points and limit cycles, continuation of branches of these invariant features, and the delineation of resonance horns for forced oscillations.

It is our belief that such a coarse-graining approach is the correct way to summarize the state of coupled heterogeneous populations, and so must play a part in any scheme for efficiently simulating large populations.
Acknowledgements

First, I would like to thank my advisor, Professor Yannis G. Kevrekidis, for the guidance and inspiration he has continuously provided me with during my time at Princeton. Yannis’s enthusiasm and deep insight are well known throughout his far-reaching network of collaborators, as is his generosity at group dinners!

I would also like to thank Professor Stanislav Shvartsman for inspiring me with the potential of biological systems for dynamical systems research, and for managing to read and provide feedback on my dissertation with such an amazingly quick turnaround. In addition, I’d like to thank my other committee members, Professors Alan Gelperin, Sankaran Sundaresan, and Jared Toettcher. Their participation has made my committee a truly excellent interdisciplinary team of advisors.

I am grateful for funding from the National Institutes of Health, the National Science Foundation’s Graduate Research Fellowships Program and a fellowship from the Princeton Program in Plasma Science and Technology. The generous support from these agencies has greatly helped me to advance my work.

In addition, I am thankful that I have had the good fortune to work with many brilliant colleagues during my time in graduate school—Prof. William Gear, Thomas Thiem, and Yan Wu at Princeton, as well as Prof. Carlo Laing of Massey University, New Zealand; Prof. Minseok Choi of Pohang University of Science and Technology, Korea; and the Felixes Kemeth and Dietrich, Sindre Haugland, and Prof. Katharina Krischer at the Technical University of Munich.

I also want to thank my friends Alexander, Dmitry, Will, Yile, and Yogesh, who shared with me the road to Ph.D. and made it so much fun. Guys—you’ll always know how much I really mean when I say of a sunset, “it’s orange”.

My special thanks go to my parents, Becky and John, and my sister Mia for their endless love and support. Your encouragement helped me make the right choice to
pursue graduate school and thanks to your clarity I have always known I would be here eventually.

Finally, I want to thank Angelina, who has been the greatest source of inspiration and joy in my life. Together, we’ve now written *two* dissertations!
Contents

Abstract ................................................................. iii
Acknowledgements .................................................... iv
List of Tables ........................................................... viii
List of Figures ........................................................... ix

1 Introduction ......................................................... 1

2 Background and methods ........................................... 5
   2.1 Coarse graining ..................................................... 6
   2.2 Orthogonal polynomial chaos expansion ....................... 10
   2.3 Diffusion maps ....................................................... 21

3 Polynomial chaos expansion for intrinsic and structural hetero-
genieties ............................................................... 24
   3.1 Introduction .......................................................... 25
   3.2 An illustrative example of heterogeneous coupled oscillator networks . 30
   3.3 Low-dimensional representation .................................. 33
   3.4 Coarse computational modeling tasks ............................ 38
   3.5 Discussion .......................................................... 45

4 Analysis of variance for intrusive polynomial chaos expansion .... 48
   4.1 Intrusive PCE for dynamics ...................................... 48
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Frequently encountered probability distributions and the corresponding weighted orthogonal polynomial families.</td>
<td>12</td>
</tr>
<tr>
<td>4.1</td>
<td>Comparison of numbers of points for sparse grids and ANOVA</td>
<td>60</td>
</tr>
</tbody>
</table>
## List of Figures

2.1 A simple heterogeneous coupled phase-oscillator system .................. 7
2.2 “Swiss roll” demonstration of diffusion maps ................................. 21

3.1 Visualization of a Chung-Lu network ........................................... 31
3.2 Oscillator states (phases) quickly slave to oscillator “identities” ........ 32
3.3 Orthogonal 2D Polynomials ....................................................... 36
3.4 Coarse projective integration ...................................................... 39
3.5 As the inner step size decreases, the error of a projective integration  
becomes bounded by the integrator’s intrinsic step size ....................... 40
3.6 Convergence of coarse fixed point with increasing basis size; convergence  
of generated polynomials with increasing population size ..................... 41
3.7 Comparison of coarse and fine eigencomputations for different basis  
sizes close to and far away from the main SNIPER bifurcation .............. 43
3.8 Bifurcation diagram, coarse and fine ........................................... 44

4.1 Sparse grids vs. ANOVA simulations of pre-Bötzingener neurons ....... 59
4.2 The mean and variance for $V$ ..................................................... 61
4.3 The mean and variance for $h$ ..................................................... 61
4.4 Coarse projective integration and detailed (fine) coupled dynamics for  
$V$ and $h$ .................................................................................. 63
4.5 Periodic orbit of the mean of $V$ and the mean of $h$ ......................... 64
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.6</td>
<td>Eigenvalues of the Jacobian of the fine flow map and the coarse flow map at corresponding fixed points, obtained with three different PCE orders</td>
<td>65</td>
</tr>
<tr>
<td>4.7</td>
<td>Detailed PCE coarsening of pre-Bötzinger system states with structural heterogeneity</td>
<td>67</td>
</tr>
<tr>
<td>4.8</td>
<td>Limit cycle of coarse pre-Bötzinger system states with structural heterogeneity</td>
<td>68</td>
</tr>
<tr>
<td>5.1</td>
<td>Use of diffusion maps to unshuffle a shuffled recording of a 1D Ginzburg-Landau PDE simulation</td>
<td>72</td>
</tr>
<tr>
<td>5.2</td>
<td>Use of diffusion maps to unshuffle a shuffled recording of a 2D Ginzburg-Landau PDE simulation</td>
<td>73</td>
</tr>
<tr>
<td>5.3</td>
<td>Use of diffusion maps to rediscover one intrinsic heterogeneity one structural from a recording of a simulation of pre-Bötzinger neurons</td>
<td>77</td>
</tr>
<tr>
<td>5.4</td>
<td>Tuning the diffusion map $\epsilon$ generates parameterizations that emphasize different-scale features of the data.</td>
<td>78</td>
</tr>
<tr>
<td>5.5</td>
<td>Parameterization of a torus of states through diffusion maps on a dataset of overlapping patches of PDE trajectory data</td>
<td>87</td>
</tr>
<tr>
<td>5.6</td>
<td>Rediscovery of a map of Germany using vectors of sunrise and sunset times over a typical year for several cities</td>
<td>88</td>
</tr>
<tr>
<td>6.1</td>
<td>Limit cycle for a circadian oscillator system of two neurons forced by a sigmoidally varying light signal</td>
<td>99</td>
</tr>
<tr>
<td>6.2</td>
<td>Isola of stable and unstable stroboscopic map solutions, for several forcing periods at a fixed forcing amplitude</td>
<td>100</td>
</tr>
<tr>
<td>6.3</td>
<td>Approximate Arnol’d tongue/resonance horn constructed from several isolas at different forcing amplitudes</td>
<td>102</td>
</tr>
</tbody>
</table>
6.4 Circadian bioluminescence data, divided into desynchronized and resynchronized regions, and compared to analogous coupled phase oscillator data .............................................. 103

6.5 Diffusion maps applied to experimental circadian bioluminescence recordings uncovers a phase $\times$ amplitude parameterization. ............. 104

6.6 Diffusion maps applied to snippets of experimental circadian bioluminescence recordings uncovers one increasing/decreasing parameterization and one concave-up/concave-down parameterization. ............. 104

6.7 Circadian simulation data and PCE fit .................................................. 106

6.8 PCE fit coefficients and error over time for SCN simulation data ........ 107
Chapter 1

Introduction

Model reduction for dynamical systems has been an important research direction for decades; accurate reduced models are very useful, and often indispensable for the understanding, analysis, and ultimately for the design of large/complex dynamical systems. The relevant tools and techniques range from center manifold reduction close to bifurcation points [42] to singular perturbation techniques (analytical [9] or computational [54]) and (more recently) to data-driven reduction methods (like PCA [47], or nonlinear manifold learning techniques [22, 28]). While many such tools are well established for ODEs and PDEs, the dynamics of networked dynamical systems (e.g. [78]) pose additional challenges.

We are interested here in large sets of dynamic units (agents, oscillators, cells) linked in a prescribed (and, for this dissertation, fixed) coupling pattern. Every unit consists here of a (relatively small) set of ordinary differential equations. The units are intrinsically heterogeneous, meaning that the parameters of this set of ODEs are

---

This work was supported by a grant from the National Institutes of Health, as well as a fellowship from the National Science Foundation. Portions appeared in publication in Frontiers in Computational Neuroscience, European Physical Journal, as well as at several conferences, including the Network Frontier workshop at Northwestern University, the Advances in Discrete Networks workshop at the University of Pittsburgh, Dynamics Days at the University of Maryland, several meetings of the American Institute of Chemical Engineers.
sampled from a probability distribution. Once the overall system of ODEs modeling a large network is assembled, any generic dynamical system model reduction technique can be tried. For all-to-all coupled heterogeneous units, in particular, there has been extensive work taking advantage of the overall model structure, leading to the systematic reduction of such intrinsically heterogeneous assemblies [63, 80, 81].

“Dynamical systems” in general can describe discrete systems, or timesteppers, ordinary differential equations (ODEs), and partial differential equations (PDEs). For the most part, this dissertation is focused on ODEs. However, a central assumption of the work done here is that many of the systems examined could potentially admit an alternate description as a PDE. In chapter 4, this alternate description is made explicit. On the other hand, the finite time flow maps defined in chapter 2 and used most extensively in chapter 6 (though also in chapter 3) are a form of discrete dynamical system.

These systems all have in common their status as descriptions for the evolution rules of a system state. We distinguish between the dynamic dimension $n_{\text{fine}}$ which is the number of quantities in this state vector, and the spatial dimension $d_u$. For the systems described here, the spatial dimension is the number of heterogeneous parameters across which the population of units vary. This may be a physical space, a heterogeneous parameter space, or a product of these. In our systems, where we keep track of $n_{\text{vars}}$ dynamic variables per unit and $n_{\text{nodes}}$ is the number of units in the population, $n_{\text{fine}}$ is usually $n_{\text{vars}}n_{\text{nodes}}$.

This dissertation is divided into chapter 2, in which we establish a common notation and describe some methods used repeatedly, and the remaining chapters, in which we apply the methods of chapter 2 to a succession of increasingly complex dynamical systems. Chapter 2 begins with the notation for and some concepts behind coarse graining. This is the computational technique by which we approximated the $n_{\text{fine}}$-dimensional state of a dynamical system with $n_{\text{coarse}} \ll n_{\text{fine}}$ quantities. We discuss
some sample computational tasks which are aided by this approach, including the solution of initial value problems and the computation of fixed points or steady states.

We continue with a discussion of polynomial chaos expansion (PCE), the Fourier-like coarse-graining method used throughout this dissertation. We additionally discuss the problem posed when $n_{\text{vars}} > 1$ and the multidimensional distribution of the heterogeneities is not separable. We propose a method for generating a PCE basis which is still orthogonal under these conditions.

Finally, we discuss in the methods discussion of chapter 2 diffusion maps, a technique for dimension reduction (unsupervised learning) in the machine learning sense. Diffusion maps can be likened to principal components analysis (PCA) in that both can be used to find a more parsimonious description for a point cloud through dimension reduction. However, unlike PCA, diffusion maps is capable of discovering nonlinear submanifolds along which the data lie. For this dissertation, diffusion maps is employed in chapter 5 to infer a heterogeneous space from recordings of system dynamics.

In chapter 3, we examine coupled phase oscillators; this is the simplest dynamical system to which we can apply our PCE-based coarse graining. Here, $n_{\text{vars}} = 1$. A simpler version of this system with only $d_u = 1$ is shown as an aside in §2.1. However, for this chapter, $d_u = 2$. The extra heterogeneity is “structural”, in the sense that it arises not from qualities intrinsic to individual oscillators, but instead from the coupling network between oscillators. In addition to introducing this structural/intrinsic heterogeneity distinction, We look at the use of a PCE coarse-graining for computing fixed points and continuing branches thereof.

In chapter 4, we upgrade our system of interest to a more biophysically realistic Hodgkin-Huxley like system, with $n_{\text{vars}} = 2$. Up to this point it was taken as implied that, though the system dynamics were described by ODEs, an underlying smoothness of system states implied that a PDE description at least made sense conceptually.
Here, this is made explicit, by writing the system dynamics as PDEs, and then using
the orthogonality of our PCE basis to compute coarse variables directly. This setting
allows us to repurpose analysis of variance to perform the expansion integrals, by which
the potentially high-dimensional integrals performed are approximated by a number
of lower-dimensional terms. The result is a method for computing the orthogonal
expansions which is less sensitive to the spatial dimension $d_u$.

In chapter 5, we employ diffusion maps, as described in chapter 2, to extract the
$d_u$ heterogeneities directly from recordings of dynamics. Some notes are also included
on tuning the diffusion maps scale parameter $\epsilon$, and on the conceptual meaning of
this parameter.

Finally, in chapter 6 we apply these methods to a realistic model of circadian
rhythms in suprachiasmatic nucleus (SCN) of the mammalian hypothalamus. The
system in question produces self-sustained oscillations with a natural period of about
one day through feedback between several mRNAs, proteins, and protein-protein
complexes. Signaling between cells uses both vasoactive intestinal peptide and $\gamma$-
aminobutyric acid. As in chapter 3 and chapter 4, we find that neural state (protein,
mRNA, etc. concentration within the cytoplasm and nucleus of a cell) becomes a
smooth function of heterogeneous neuron parameters (such as base transcription
rates of some mRNAs). The “equal space” method of chapter 5 is applied to experi-
mental data, demonstrating that the population of neurons can be embedded in a
two-dimensional space best described as the span of a “phase” heterogeneity and
an “amplitude” heterogeneity. Automated methods for finding limit cycles of the
unforced dynamics are employed, as well as methods for finding limit cycles under a
regular external forcing signal. For the forced dynamics, a resonance horn diagram is
constructed, in which both the period and the amplitude of the external forcing signal
are varied.
Chapter 2

Background and methods

In this chapter, we detail the methods used multiple times in the remaining chapters of this dissertation. The central topic of this dissertation is dynamical systems, and methods for dimension reduction in dynamical systems.

First, we discuss the concept of coarse graining for dynamical systems work, in which we make use of our ability to represent the state of a system more parsimoniously to accelerate computational tasks with that system. These computational tasks include the solution of initial value problems; the discovery of fixed points, limit cycles, or other invariants of the dynamics; the description of these invariants as stable, unstable, or neutrally stable; and the use of continuation of these invariants to produce bifurcation diagrams.

Second, we discuss polynomial chaos expansion (PCE), which is the dimensionality reduction method of choice for the classes of dynamical systems of interest for this dissertation. Here, we note some basic results for orthogonal projection, show that orthogonality of a particular choice of multidimensional basis holds when an independence condition is true, and outline an approach for generating a basis when this condition is violated.
Finally, we discuss diffusion maps. Confusingly, while this is itself a dimensionality reduction technique, here we use it not to reduce the dynamic dimension of the system, but the spatial dimension for the underlying PDE; a distinction that will be clarified below. That is, diffusion maps is used as a stepping stone to allow for effective use of polynomial chaos expansion.

2.1 Coarse graining

With the use of methods such as the polynomial chaos expansion of §2.2 and the diffusion maps of §2.3, we can often describe high-dimensional data using a much-reduced set of coarse variables. An example of the sort of dimension reduction developed in this dissertation is hinted at by Fig. 2.1. Here, we compute a trajectory for a simpler analog to the dynamical system treated in more depth in chapter 3. It can be seen in Fig. 2.1(d) that, after a short transient, the complicated (and un-summarizable) initial state is replaced with a simple (and summarizable) system state. It is this summarization that we seek to use for computational tasks— the full portrait of Fig. 2.1(d) (all the \( \theta \) values) is referred to as the fine variables, and our summarization, whatever form that might take, is referred to as the coarse variables.

Beyond their conceptual simplification value, collective (coarse) variables can be valuable in facilitating the computer-assisted study of complex dynamical systems by accelerating tasks such as direct simulation, continuation, stability and bifurcation analysis for different types of solutions. To accomplish this acceleration, the equation-free approach [56, 55] is predicated on the ability to map between corresponding fine and coarse descriptions of the same system.

This is accomplished through the definition of a restriction operator \( R : \mathbb{R}^{n_{\text{fine}}} \to \mathbb{R}^{n_{\text{coarse}}} \) which maps from fine states \( \theta(t) \) to corresponding coarse states \( \alpha(t) \). We also
Figure 2.1: State of dynamical system $d\theta_i/dt = \omega_i + (K/n_{\text{nodes}}) \sum_{i=1}^{n_{\text{nodes}}} \sin(\theta_j - \theta_i)$ at three points in time. Initial conditions $\theta_i(t = 0)$ and natural frequencies $\omega_i$ are uniformly distributed as shown in 2.1(b).

need to define the counterpart of restriction: a lifting operator $L : \mathbb{R}^{n_{\text{coarse}}} \rightarrow \mathbb{R}^{n_{\text{fine}}}$ which maps $\alpha$ vectors to $\theta$ vectors.

One more important thing to note is that the lifting operator is, in general, a one-to-many relation; there are many fine realizations of the process that are mapped to the same coarse representation—coarse-graining (e.g. averaging) loses information. If the problem can be usefully coarse-grained, any of these consistent fine realizations, or the average of several of them, can be used practically in the definition of the coarse time-stepper below; we may think of the coarse-timestepper as the expected value over all such consistent realizations. In singularly perturbed multiscale problems one can clearly see how the memory of the details of the lifting are quickly forgotten, suggesting that any consistent fine realization is “good enough” to estimate this expectation [39, 57].
The $L$ and $R$ operators combine to define a coarse timestepper $\Phi_{\tau,C}$, in

$$
\Phi_{\tau,C} : \mathbb{R}^{n_{\text{coarse}}} \rightarrow \mathbb{R}^{n_{\text{coarse}}}
$$

$$
\Phi_{\tau,F} : \mathbb{R}^{n_{\text{fine}}} \rightarrow \mathbb{R}^{n_{\text{fine}}}
$$

$$
\Phi_{\tau,F}[\theta(t)] = \theta(t + \tau) = \int_{s=t}^{s=t+\tau} \frac{d\theta(s)}{dt} ds
$$

$$
\Phi_{\tau,C}[\alpha(t)] = \alpha(t + \tau) \equiv (R \circ \Phi_{\tau,F} \circ L)[\alpha(t)]
$$

(2.1)

This is the timestepper for the (unavailable) coarse-grained dynamical system, approximated through observing the results of short bursts of appropriately initialized fine-grained simulations. A single evaluation of this coarse time-stepper, by itself, does not provide any computational savings; it is the way we design, and process the results of, several such coarse time-steps that leads to computational benefits. Using traditional numerical analysis codes (initial value solvers, fixed point solvers) as templates for wrapper codes around the coarse timestepper, tasks like accelerated simulation, coarse-grained stability and bifurcation analysis, optimization, and controller design, can be performed. This wrapper technology is described in detail (and fruitfully used to explore model coarse-graining across disciplines) in a series of publications \cite{109, 58}. What is important here is not the established wrapper algorithms technology; it is the selection of coarse observables, leading to the appropriate definition of the coarse time-stepper, that makes the entire program feasible and useful.

**Coarse initial value problems**

We can use the coarse timestepper to accelerate the computation of dynamic trajectories of the system, through Coarse Projective Integration (CPI) \cite{37, 66}. Given a coarse initial condition $\alpha(t = 0)$ we lift to a consistent fine scale state $L[\alpha(t = 0)]$ and use it to initialize a fine scale numerical integrator. We run for a short time $\tau$ (the inner
step) and we record the final coarse state by restricting the corresponding fine state, $R[\theta(t + \tau)]$. We use these two coarse states to estimate the coarse time derivative, which we then use in the forward Euler formula to project forward in time the coarse state for a (large, coarse) time step $h$ (the outer step). This constitutes the simplest coarse projective forward Euler integration scheme:

$$\alpha(t + h) = \alpha(t) + h \frac{\Phi_{\tau,C}[\alpha(t)] - \alpha(t)}{\tau} \quad (2.2)$$

A slightly more sophisticated approach would also take two points separated by an inner step size $\tau$ to approximate the rate-of-change of $\alpha$, but only after first performing a healing integration in the fine equations [39]. If the lifted representation of the state $\alpha(t)$, as projected from the previous timestep $\alpha(t - h)$, is slightly off the hypothetical slow manifold in the fine space, this short healing trajectory dampens the fast components which are not captured in the coarse representation [111, 6, 36]. Such an approach is more appropriate for stochastic dynamics than for the deterministic (though heterogeneous) systems described in this dissertation.

Results of applying the simpler scheme to a coupled oscillator problem (described in detail in chapter 3) are shown in Fig. 3.4. In general, the coarse time-stepper results can be used to estimate the coarse ODEs $d\alpha/dt$, which are not available in closed form. On demand numerical estimates of this right-hand-side through short bursts of appropriately initialized fine simulation allow us to use other existing integrators, such as scipy.integrate.odeint, to computationally approximate coarse trajectories (also shown in Fig. 3.4). Appendix 2 of [10] contains a quick illustration of the useful properties of such projective initial value solvers. It is shown there that, under reasonable conditions, the order of a projective integrator templated on a two-step Runge-Kutta initial value solver (including the additional estimation step for the coarse time derivatives) is the same as the order of the actual Runge-Kutta initial value
solver. Figure 9 in [10] confirms this for projective integration of the fine equations for our model.

This approximation of the coarse right-hand-side function can be used for other computational tasks besides the computation of dynamic trajectories, such as the discovery of fixed points or limit cycles, and the continuation of these features with changing system parameters.

**Coarse fixed point computation**

The coarse time-stepper can be used to define a coarse difference map

\[
F_\tau[\alpha(t)] = \Phi_{\tau,C}[\alpha(t)] - \alpha(t). \tag{2.3}
\]

Steady states of the fine time stepper are clearly zeros of this difference; one expects that the zeros of the coarse difference may correspond to coarse steady states of the original problem (when they aren’t actually time-\(\tau\) periodic points, as described in §6.2). \(F_\tau\) can therefore be used to find coarse steady states involving only \(n_{\text{coarse}}\) variables. Iterative, matrix-free linear algebra lends itself to finding zeros of such a problem in the absence of explicit equations for the dynamics of the coarse variables \(\alpha_j\).

In chapter 3, we used a Krylov-type matrix free technique (Newton-Krylov GMRES) to converge to such coarse steady states.

### 2.2 Orthogonal polynomial chaos expansion

We now discuss a particular coarse-graining method applicable to systems such as that depicted in Fig. 2.1(d).

While the example of Fig. 2.1 can be described as a set of ODEs

\[
\frac{dY(X, t)}{dt} = F(X, Y, t) \tag{2.4}
\]
where \( Y = \{ \theta_i^{(j)} \}_{i=1, \ldots, n_{\text{nodes}}; j=1, \ldots, n_{\text{vars}}} \) comprises all the dynamic variables over all the units, and \( X = \{ \xi_i^{(k)} \}_{i=1, \ldots, n_{\text{nodes}}; k=1, \ldots, d_u} \) comprises all the heterogeneities.

Alternately, due to the smoothness visible in Fig. 2.1(d), we might hypothesize that Eqn. (2.4) could be expressed as a PDE, as

\[
\frac{\partial}{\partial t} \theta(\xi; t) = \begin{bmatrix}
\frac{\partial}{\partial t} \theta^{(1)}(\xi^{(1)}, \ldots, \xi^{(d_u)}; t) \\
\vdots \\
\frac{\partial}{\partial t} \theta^{(n_{\text{vars}})}(\xi^{(1)}, \ldots, \xi^{(d_u)}; t)
\end{bmatrix} = f(\theta(\xi; t), \xi, t),
\]

where the dynamic variables \( \theta \) are no longer indexed over units \( i = 1, \ldots, n_{\text{nodes}} \), but instead are assumed to be smooth functions of the heterogeneities \( \xi = \{ \xi^{(1)}, \ldots, \xi^{(d_u)} \} \).

For the remaining exposition in this section, we restrict ourselves to \( n_{\text{vars}} = 1 \) and \( d_u = 1 \) for simplicity, obviating the need for bracketed superscripts, and leaving only subscripts over units or over basis functions.

Since the states we wish to coarse-grain are predictor-response datasets, standard regression techniques, such as polynomial regression, are applicable. In such an approach, we would express the smooth functional dependence \( \theta(\xi) \) as a low-order polynomial. That is, a truncated linear combination

\[
\theta(\xi, t) \approx \sum_{k=1}^{n_{\text{coeffs}}} \alpha_k(t)\psi_k(\xi)
\]

of the basis functions \( \psi_k(\xi) = \xi^{k-1} \).

However, for reasons that will become apparent in the next subsection, we will impose the extra requirement that our basis functions be orthogonal under a particular inner product. The real importance of this will be shown in chapter 4, where we examine the interaction of the PDE of Eqn. (2.5) with the approximation of Eqn. (2.6).

For our problems, we can construct this inner product as an expectation and satisfy this requirement when \( \xi \) is a random variable, and \( \theta \) is a function of \( \xi \), possibly
corrupted by noise. In this case, a good scheme for approximating $\theta(\xi)$ is the Wiener-Askey polynomial chaos method. [119, 41] This technique is motivated by the use of sequences of orthogonal polynomials in the field of uncertainty quantification (UQ), where they are used to decouple systems of equations across stochastic dimensions, in a manner similar to the use of Fourier modes for systems defined across space or time. Some standard distributions and their corresponding families of orthogonal polynomials [119] are given in Table 2.1.

<table>
<thead>
<tr>
<th>Discrete Distribution</th>
<th>Discrete Polynomials</th>
<th>Continuous Distribution</th>
<th>Continuous Polynomials</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binomial</td>
<td>Kravchuk</td>
<td>Gaussian</td>
<td>Hermite</td>
</tr>
<tr>
<td>Poisson</td>
<td>Charlier</td>
<td>Gamma</td>
<td>Laguerre</td>
</tr>
<tr>
<td>Negative binomial</td>
<td>Meixner</td>
<td>Beta</td>
<td>Jacobi</td>
</tr>
<tr>
<td>Hypergeometric</td>
<td>Hahn</td>
<td>Uniform</td>
<td>Legendre</td>
</tr>
</tbody>
</table>

Table 2.1: Frequently encountered probability distributions and the corresponding weighted orthogonal polynomial families.

**Calculation of fitting coefficients by inner product**

The principal method for expressing the fitting coefficients $\alpha_i$ is a Galerkin projection as expressed as a Lebesgue inner products.

$$
\alpha_k(t) = \langle \theta(\xi, t), \psi_k(\xi) \rangle_\rho \overset{def}{=} \int_{\xi \in \Omega} \theta(\xi, t) \psi_k(\xi) d\Gamma(\xi) 
$$

(2.7)

In order to show that this method of projection holds, we rely on orthogonality of the chosen basis functions under the weighting again expressed as an inner product. That is,

$$
\delta_{jk} = \langle \psi_j(\xi), \psi_k(\xi) \rangle_\rho = \mathbb{E}_\rho [\psi_j(\xi) \psi_k(\xi)] ,
$$

(2.8)

where, again, the interpretation of this integral as an expectation comes into play when the space $\Omega$ is one of heterogeneous (random) parameters, and the weighting $\rho$ is a probability density.
Taking the inner product between basis function and the expansion of our target function, we can see that, by orthogonality, this gives us the coefficients $\alpha$ of our expansion Eqn. (2.6).

$$\langle \theta, \psi_k \rangle \approx \left\langle \sum_j \alpha_j \psi_j, \psi_k \right\rangle \rho$$

$$= \int_\Omega \left[ \sum_j \alpha_j \psi_j(\xi) \psi_k(\xi) \right] d\Gamma(\xi)$$

$$= \sum_j \alpha_j \langle \psi_j(\xi), \psi_k(\xi) \rangle _\rho$$

$$= \sum_j \alpha_j \delta_{jk}$$

$$= \alpha_k$$

Thus, we obtain the standard Galerkin projection result of Eqn. (2.7)–coefficients can be obtained through the inner product between the function to be approximated and each basis function.

**Calculation of fitting coefficients by least squares**

For sampled $(\xi_i, \theta_i)$ data, or situations when it is impractical to compute the integral of Eqn. (2.7), an alternative approach makes use of the interpretation of this inner product as an expectation, by using the sample mean as an approximation. This is exactly equivalent to minimizing the mean squared error of the reconstruction of Eqn. (2.6), and can be expressed concretely as a linear problem

$$A_{i,j} \hat{\alpha}_j \approx \theta_i$$

$$A_{i,j} = \psi_j(\xi_i) / \sqrt{\sum_l \psi_j(\xi_l)^2}$$

$$\alpha_j = \hat{\alpha}_j / \sqrt{\sum_l \psi_j(\xi_l)^2}$$

(2.10)
In Eqn. (2.11), the divisor $\sqrt{\sum_i \psi_j(\xi_i)^2}$ is present to improve the condition number of the Vandermonde matrix $A$. This is the approach used in `numpy.polynomial.hermite.hermval` [2, 1], and is discussed in more detail in §3.3.

**Choice of basis**

The *polynomial chaos expansion (PCE)* [121, 70, 11] approximates $\theta(\xi)$ as a linear combination of a set of orthogonal basis functions, $\psi_k(\xi)$ which are chosen such that the weighting with respect to which they are orthogonal is similar to the actual distribution of the $\xi$ values. For instances, if the distribution of $\xi$ is uniform, Legendre polynomials are indicated; and if it is Gaussian, Hermite polynomials, the first few of which\(^1\) are $H_0(\xi) = 1$, $H_1(\xi) = \xi$, $X_2(\xi) = \xi^2 - 1$, $H_3(\xi) = \xi^3 - 3\xi$, and $H_4(\xi) = \xi^4 - 6\xi^2 + 3$. Further polynomials can be generated by a recurrence relation [119].

This situation can be further complicated by adding more heterogeneities, or by considering systems in which there is both an intrinsic heterogeneity, associated with specific individuals, and a structural heterogeneity which arises from the individuals’ interactions (The inclusion of such a structural heterogeneity makes this a network as well as an ODE problem.) Here, the function to be fit will be not a curve but a hypersurface.

One way we might generalize to the fitting of any dimension of hypersurface by constructing multidimensional polynomials from the product of one-dimensional polynomials, (e.g. $G_{i,j}(\xi_1, \xi_2) = \psi_i(\xi_1)\psi_j(\xi_2)$). When a basis of polynomial order $k$ is requested, we construct all the products whose cumulative order is less than or equal to $k$. (I.e., $\{\psi_i(\xi_1)\psi_j(\xi_2) \forall (i,j) \in \mathbb{N}^2 | i + j \leq k\}$ So, for $k = 3$, there are 10 basis functions, and so the number of coefficients in the fit, and thus the number of coarse variables, will also be 10.)

\(^1\)These are the “probabilist’s” Hermite polynomials, which should be distinguished from the slightly different “physicist’s” Hermite polynomials.
The least-squares technique described in Eqn. (2.10) works equally well for basis functions of $d_u$ variables. The $n_{\text{nodes}}$ formerly scalar elements of $x_i$ each become vectors of length $d_u$, and the arities of the basis functions correspondingly increase from 1 to $d_u$. The shape of the Vandermonde matrix $A$ remains $(m, n)$, where $m$ is the number of units in the population, and $n$ is the number of functions in the basis. So, the cost of creating the Vandermonde matrix increases, but the cost of the required least-squares solution remains the same.\(^2\)

**Orthogonality of the tensor product basis for independent weightings**

Here, we demonstrate that if we have polynomials in two variables that are orthogonal with respect to a two-dimensional weight function that is the product of two one-dimensional weight functions, then the orthogonal polynomials are the product of the respective one-dimensional orthogonal sets.

We suppose that we are given two one-dimensional orthogonal function sets defined on (possibly infinite) intervals $I_x$ and $I_y$ based on the weight functions $\rho(x)$, $x \in I_x$, and $\sigma(y)$, $y \in I_y$ and we want to construct a set of two-variable orthonormal polynomials, $\psi_{i,j}(x,y)$ of degrees $i$ in $x$ and $j$ in $y$ such that

$$\langle \psi_{i,j}, \psi_{m,n} \rangle_{\rho \sigma} = \delta_{im} \delta_{jn}$$ (2.12)

where

$$\langle f, g \rangle_{\rho \sigma} = \int\int f(x,y)g(x,y)\rho(x)\sigma(y)dxdy$$ (2.13)

\(^2\)If the expansion is to be performed multiple times over the course of a larger computation, it may be beneficial to cache the Vandermonde matrix for repeated reuse, since it depends only on the heterogeneous parameters, which are assumed to be constant, and on the definitions of the basis functions, which themselves depend on the distributions of the heterogeneities.
and \( \{\psi_i(x)\} \) and \( \{\theta_j(y)\} \) are the orthonormal sets of polynomials corresponding to the one-dimensional systems based on the weights \( \rho(x) \) and \( \sigma(y) \) respectively, that is, \( \langle \psi_i, \psi_j \rangle_\rho = \delta_{ij} \) and \( \langle \theta_i, \theta_j \rangle_\sigma = \delta_{ij} \). We will see that these restrictions lead to the product polynomials \( \psi_{i,j}(x,y) = \psi_i(x)\theta_j(y) \).

Since \( \psi_{i,j}(x,y) \) has maximum degree of \( i \) in \( x \) and \( j \) in \( y \), it can be written as

\[
\psi_{i,j}(x,y) = \sum_{p \leq i, q \leq j} A_{ijpq} \phi_p(x)\theta_q(y) \tag{2.14}
\]

Substituting Eqn. (2.14) in Eqn. (2.12) we get

\[
\delta_{im}\delta_{jn} = \sum_{p \leq i, q \leq j} \sum_{r \leq m, s \leq n} A_{ijpq} A_{mnrs} \langle \phi_p(x), \phi_r(x) \rangle_\rho \langle \theta_q(y), \theta_s(y) \rangle_\sigma = \sum_{p \leq i, q \leq j} \sum_{r \leq m, s \leq n} A_{ijpq} A_{mnrs} \delta_{pr} \delta_{qs} = \sum_{p \leq \min(i,m), q \leq \min(j,n)} A_{ijpq} A_{mnpq} \tag{2.15}
\]

We prove the result by induction on \( k = m + n \) showing that \( A_{ijmn} = \delta_{im} \delta_{jn} \). For \( k = 0 \) from Eqn. (2.15), we have immediately that \( A_{0000} = 1 \) and \( A_{ij00} A_{0000} = 0 \) for \( i + j > 0 \). Since the signs of the one-dimensional orthogonal polynomials are arbitrary, we can take \( A_{0000} = 1 \). The second relations implies that \( A_{ij00} = 0 \) for \( i + j > 0 \). This establishes the result for \( k = 0 \).

Assuming that it is true for \( k - 1 \), we examine Eqn. (2.15) with \( m + n = k \). Setting \( i = m \) and \( j = n \) in Eqn. (2.15) we have \( A_{mnnn}^2 = 1 \) allowing us to choose \( A_{mnnn} = 1 \). Then, for any \( (i,j) \neq (m,n) \) we have \( A_{ijmn} A_{mnnn} = 0 \) implying that \( A_{ijmn} = 0 \), thus establishing the result.
Multidimensional polynomial generation by algebraic Gram-Schmidt

When producing a set of orthogonal basis functions \( \{\psi_k(\xi_1, \ldots, \xi_d)\}_{k=1}^{n_{\text{coeffs}}} \), it is necessary to choose the functions such that the orthogonality of Eqn. (2.12) is maintained. For example, in chapter 3, the domain variables are \( \xi_1 = \omega \) and \( \xi_2 = \kappa \). Because the weighting function \( \rho(\omega, \kappa) = \rho_\omega(\omega)\rho_\kappa(\kappa) \) is the product of unidimensional weighting functions, the orthogonality integral of Eqn. (2.13) can be separated into two 1D integrals, and a product basis of the type discussed above can be used.

Alternately, we can devise a scheme by which we generate orthogonal polynomials directly from the moments (whether analytic or empirical) of the distribution. To derive this scheme for multidimensional polynomials, we begin by showing how it applies in the unidimensional case.

If the \( k \)th polynomial is defined as

\[
\psi_k(x) = \sum_{i=0}^{k} p_i^{(k)} x^i, \tag{2.16}
\]

with \( \Omega \ni x \) being the support of the random variable, and \( d\Gamma(x)/dx \) being its density function, the orthogonality conditions for this polynomial can be written directly as

\[
\int_{x \in \Omega} p_0^{(0)} \sum_{i=0}^{k} p_i^{(k)} x^i \ d\Gamma(x) = 0
\]

\[
\int_{x \in \Omega} \left[ \sum_{i=0}^{k} p_i^{(k)} x^i \right] \left[ \sum_{i=0}^{k} p_i^{(k)} x^i \right] d\Gamma(x) = 0
\]

\[
\int_{x \in \Omega} \left[ \sum_{i=0}^{k-1} p_i^{(k-1)} x^i \right] \left[ \sum_{i=0}^{k} p_i^{(k)} x^i \right] d\Gamma(x) = 0
\]

\[
p_k^{(k)} = 1,
\]

(2.17)

The methods described in this subsection are discussed in more depth in [24], a senior thesis the author of which was mentored by the author of this dissertation. Here, we reproduce the central argument.
where the final line reflects the further requirement that the polynomials should be monic (their highest-order term should have coefficient 1). This serves as an easier-to-express alternative to requiring that the polynomials be normalized (in which case the inner product of a polynomial with itself is one). When using such non-normalized polynomials, it is necessary in use to divide each by the normalizing constant \( \sqrt{\langle \psi_k(x), \psi_k(x) \rangle}_\rho \), as in Eqn. (3.11).

We can rearrange as in [79] in terms of the raw moments \( \mu_k = \int_\Omega x^k d\Gamma(x) \),

\[
\begin{pmatrix}
\mu_0 & \mu_1 & \cdots & \mu_k \\
\mu_1 & \mu_2 & \cdots & \mu_{k+1} \\
\vdots & \vdots & \ddots & \vdots \\
\mu_{k-1} & \mu_k & \cdots & \mu_{2k-1} \\
0 & 0 & \cdots & 1
\end{pmatrix}
\begin{pmatrix}
p_0^{(k)} \\
p_1^{(k)} \\
\vdots \\
p_{k-1}^{(k)} \\
p_k^{(k)}
\end{pmatrix}
= \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}.
\]

Since the result is written in terms of moments, this method can be used with both discrete and continuous variables, and can be applied to sampled data by using sample moments.

Since the matrix of moments can involve large powers of sampled data, it can quickly become poorly conditioned. As such, we seek a solution to Eqn. (2.18) by the singular value decomposition (SVD)-based approach described in chapter 2 of [84]. We write Eqn. (2.18) as \( \mathbf{A} \mathbf{p} = \mathbf{b} \). We then let \( \mathbf{UAV}^T = \mathbf{A} \) be the SVD of \( \mathbf{A} \), where the non-negative diagonal of \( \Lambda \) is sorted in decreasing order. We next find \( r \), the number of elements \( l \) on the diagonal of \( \Lambda \) which satisfy \( l \leq \beta \cdot \max(\Lambda) \). Finally, we approximate \( \mathbf{p} \) with \( \mathbf{V} \cdot [\hat{\mathbf{S}}^{-1} \cdot (\hat{\mathbf{U}}^T \cdot \mathbf{b})] \), where \( \hat{\mathbf{S}} \) is the first \( r \) rows and columns of \( \Lambda \), \( \hat{\mathbf{U}} \) is the first \( r \) columns of \( \mathbf{U} \), and \( \beta = 10^{-6} \) is a parameter.
Multimoments

The approach described above can be used to generate polynomials orthogonal with respect to an arbitrary unidimensional distribution. When the distribution of \((x_1, \ldots, x_{d_u})\) is separable as in Eqn. (2.13), this separability enables us to use products of 1D polynomials which are drawn from orthogonal bases chosen to match the weightings \(\rho_{x_j}(x_j)\). For the independent \((\omega, \kappa)\) distributions used in chapter 3, this was the method used.

However, with a creative change in notation, we can see that almost exactly the same approach can produce orthogonal polynomials in more than one variable, even when the weighting function is not separable.

We begin by defining a multi-index \(k = [k_1, \ldots, k_{d_u}]\), which includes in the \(k_j\) the powers of the polynomial basis function in the corresponding constituent variables \(\xi^{(j)}\). Furthermore, we assign an arbitrary total ordering to the multi-indices \(k\), such as the lexicographic ordering. We allow the value of \(k\) in the notation \(k\) (such as \(0, 1, \text{etc.}\)) to be the rank of the multi-index in this ordering. So, for instance, we might have \(0 = [0, 0], 1 = [1, 0], 2 = [0, 1], \text{etc.}\) We define integer addition, subtraction, multiplication, and comparison (greater than, less than, and equality) to operate on this underlying total order, such that \(0 + 1 = 1 = [1, 0] < [0, 1] = 2\).

With these definitional changes, it becomes possible to write equations analogous to Eqn. (2.16) through Eqn. (2.17). For instance, we can define the multidimensional polynomials in terms of multidimensional monomials, as

\[
\psi_k(x_1, \ldots, x_{d_u}) = \sum_{l=0}^{k} \left( p^{(k)}_l \prod_{j=1}^{d_u} x_j^l \right). \tag{2.19}
\]
Abbreviating the integrals in Eqn. (2.17) as expectations, we can write the orthogonality requirements as

\[
\begin{align*}
\mathbb{E} [\psi_0(x_1, \ldots, x_{d_u}) \psi_k(x_1, \ldots, x_{d_u})] &= 0 \\
\mathbb{E} [\psi_1(x_1, \ldots, x_{d_u}) \psi_k(x_1, \ldots, x_{d_u})] &= 0 \\
\mathbb{E} [\psi_{k-1}(x_1, \ldots, x_{d_u}) \psi_k(x_1, \ldots, x_{d_u})] &= 0 \\
p_k^{(k)} &= 1.
\end{align*}
\] (2.20)

We finally note that only the notation, but not the underlying definition of raw moments should be changed to create a “raw multimoment”,

\[
\mu_k = \int_{\Omega(x_1, \ldots, x_{d_u})} \left( \prod_{j=1}^{d_u} x_j^{k_j} \right) d\Gamma(x_1, \ldots, x_{d_u}) = \mathbb{E} \left[ 1 \cdot \prod_{j=1}^{d_u} x_j^{k_j} \right]
\] (2.21)

At this point, it can be shown (by analogy to the reasoning from the unidimensional case) that we can write a matrix form for these conditions almost identical to Eqn. (2.18), as

\[
\begin{bmatrix}
\mu_0 & \mu_1 & \cdots & \mu_k \\
\mu_1 & \mu_2 & \cdots & \mu_{k+1} \\
\vdots & \vdots & \ddots & \vdots \\
\mu_{k-1} & \mu_k & \cdots & \mu_{2k-1} \\
0 & 0 & \cdots & 1
\end{bmatrix}
\begin{bmatrix}
p_0^{(k)} \\
p_1^{(k)} \\
p_{k-1}^{(k)} \\
p_k^{(k)}
\end{bmatrix}
= \begin{bmatrix} 0 \\
0 \\
0 \\
1 \end{bmatrix}
\] (2.22)

For sampled data, these conditions provide a concrete scheme for computing the monomial coefficients of the \(k\)th polynomial. To generate a set of \(n_{\text{coarse}}\) orthogonal polynomials, it is necessary to compute the first \(2n_{\text{coarse}} - 1\) raw multimoments. For each polynomial \(\psi_k\), a different linear system can then be written from this pool of computed multimoments, plus the appended final line of \(k\) zeros and a one.
Figure 2.2: “Swiss roll” demonstration of diffusion maps. 2.2(a) shows the original, 3D data, colored by the datapoints’ projections onto the first significant eigenvector. A small sphere of radius $\epsilon$ is plotted around a chosen datapoint. 2.2(b) shows these colorings for the first 16 nontrivial eigenvectors (increasing harmonics of 2D sines and cosines along the plane of the unwrapped roll), excluding the trivial first eigenvector.

### 2.3 Diffusion maps

While principal components analysis (PCA) and related techniques are able to capture and compactly describe linear low-dimensional structures embedded in high-dimensional data, it fails when the structures to be discovered are not hyperplanes, but instead some sort of nonlinear submanifolds of the full high-dimensional space.

The diffusion mapping of a set of data points in $\mathbb{R}^{d_{\text{data}}}$ captures the nonlinear structure of manifolds in a given $d$-dimensional point cloud [22, 74, 99, 20, 19]. Applied properly, diffusion maps define a new parameterization of the data that will delineate both any discontinuous clusters and any smooth manifolds (e.g., of dimension lower than $d_{\text{data}}$) that may exist in the data. Like PCA, this is useful both to define new coordinates that might better describe the physics of the data than those in which it is measured, and also reduce the data to the three-or-less coordinates required for plotting.

Given a collection of $N$ datapoints, each in $d$ dimensions, the diffusion maps method proceeds by first creating a pairwise weight matrix across the datapoints,
using the heat kernel to calculate the weights. The rows are then normalized by dividing by their sums, giving a Markov matrix $A$.

$$W_{i,j} = k(d(x_i, x_j)) = \exp\left(\frac{-d^2(x_i, x_j)}{\epsilon}\right)$$

$$D_{i,i} = \sum_{j=1}^{N} W_{i,j}$$

$$D_{i\neq j} = 0$$

$$A = D^{-1}W$$

(2.23)

Being a row-stochastic matrix (the rows each sum to 1), $A$ represents transition probabilities for a Markov process on the graph represented by $W$. That is, $\phi(t) = \phi(0)A^t$ describes the evolution of the distribution of occupancy probabilities for random walkers with initial distribution $\phi(0)$ across the graph. If there is a significant gap in the spectrum of $A$—that is, there are a few leading eigenvalues which are significantly larger, by ratio, than all smaller eigenvalues—then projection onto the span of these vectors constitutes a low-dimensional reparameterization of the data. If this gap occurs within the first few eigenvalues, then this projection can be used as a reduced description of the original datapoints.

In summary, the span of the top few eigenvectors of $A$ (those with the eigenvalues of largest absolute value, or that are closest to 1) describe a manifold of typical movement of random walkers on the graph. With this, we can call the diffusion map itself the mapping

$$dmap : \mathbb{R}^{d_{data}} \rightarrow \mathbb{R}^{d_{dmap}}$$

$$dmap(x_1, \ldots, x_{d_{data}}) = (\phi_1, \ldots, \phi_{d_{dmap}}).$$

(2.24)

\footnote{Practically, this projection has already been performed after the eigenproblem has been solved, as the $i$th element of the $j$th eigenvector is the projection coefficient of the $i$th datapoint onto the $j$th graph component. If $W$ is thresholded at some level, and then all nonzero values replaced with 1, then these components can be viewed as weakly-connected subgraphs of the data’s graph, whose adjacency matrix is $W$.}
However, unlike in PCA, the ordering of the eigenvectors above the spectral gap is not necessarily important. It is possible that some eigenfunctions will be harmonics of others, in which case the projections on both should not be used together as reduced variables, since they represent the same information. For example, in Fig. 2.2(b), the sorting of eigenvectors after the first can be influenced by the length-to-width ratio of the unrolled 2D manifold. Generally, one must plot one eigenvector versus another to ascertain that they form a cloud, and not a curve. Directly checking the global correlation of the two chosen eigenvectors will not always work, since their functional dependence might be nonlinear. However, an automated approach, using a local linear regression, is described in [27]. Briefly, eigenvectors are first ordered by descending corresponding eigenvalue. Then, each eigenvector past the first is approximated by a locally-linear regression in terms of all previous eigenvectors. Those which can be approximated with low error in this way are deemed harmonics, and excluded from the set \( \phi_1, \ldots, \phi_{d_{dmap}} \). This method is used in chapter 6, e.g. in Fig. 6.5(a).

In Fig. 2.2, we show the application of diffusion maps to a cloud of points in \( \mathbb{R}^3 \). The data is constructed by beginning with a spiral in \( x \) and \( y \), extruding it along \( z \), and then adding noise. With a value of \( \epsilon \) that is slightly larger than the typical distance between nearest neighbors, the first few nontrivial eigenvectors of \( A \) recover the natural parameterization of the roll in terms of \( \sin(s) \) (where \( s \) is an arclength variable) and vertical position \( \sin(z) \). The positions are recovered as \( \sin(\cdot) \), since sines and cosines are the analytical eigenfunctions of the diffusion operator, \( \nabla^2[\cdot] \). Latter eigenvectors are harmonics of the first few \( \sin(2\cdot) \), etc.
Chapter 3

Polynomial chaos expansion for intrinsic and structural heterogeneities

Finding accurate reduced descriptions for large, complex, dynamically evolving networks is a crucial enabler to their simulation, analysis, and, ultimately, design. Here we propose and illustrate a systematic and powerful approach to obtaining good collective coarse-grained observables—variables successfully summarizing the detailed state of such networks. Finding such variables can naturally lead to successful reduced dynamic models for the networks. The main premise enabling our approach is the assumption that the behavior of a node in the network depends (after a short initial transient) on the node identity: a set of descriptors that quantify the node properties, whether intrinsic (e.g. parameters in the node evolution equations) or structural (imparted to the node by its connectivity in the particular network structure). The approach creates a natural link with modeling and “computational enabling technology”
developed in the context of uncertainty quantification. In our case, however, we will not focus on ensembles of different realizations of a problem, each with parameters randomly selected from a distribution. We will instead study many coupled heterogeneous units, each characterized by randomly assigned (heterogeneous) parameter value(s). One could then coin the term heterogeneity quantification for this approach, which we illustrate through a model dynamic network consisting of coupled oscillators with one intrinsic heterogeneity (oscillator individual frequency) and one structural heterogeneity (oscillator degree in the undirected network). The computational implementation of the approach, its shortcomings and possible extensions are also discussed.

### 3.1 Introduction

Our illustrative example is a simulation of coupled phase oscillators whose dynamics are governed by the equations

\[
\frac{d\varphi_i(t)}{dt} = \dot{\omega}_i + K \frac{1}{n_{\text{nodes}}} \sum_{j=1}^{n_{\text{nodes}}} A_{i,j} \sin(\varphi_j(t) - \varphi_i(t)),
\]

where \(i = 1, \ldots, n_{\text{nodes}}\), and \(A_{i,j} \in \{0, 1\}\) is the adjacency matrix for a network with identical edges. This model was originally formulated by Yoshiki Kuramoto with all-to-all coupling (i.e., \(A_{i,j} = 1 \forall i, j\)) [59, 60]. While we work with the simplified Kuramoto oscillator system, the methods presented here have been shown to work for more realistic coupled-oscillator systems as well, such as Hodgkin-Huxley-like neurons [16], metabolizing cells [11], gene-expression oscillations in circadian rhythms (ongoing work), or other candidate systems [7]. In cases such as the Hodgkin-Huxley, where each unit is described by multiple dynamic variables, (e.g, membrane potential and gating variables), the analysis used here is repeated for each per-cell variable.
In order to construct a frame that moves with the average phase angle, we use states $\theta \in \mathbb{R}^{n_{\text{fine}}=n_{\text{nodes}}-1}$, where
\[
\theta_i(t) \overset{\text{def}}{=} \left( \phi_i(t) - \frac{1}{n_{\text{nodes}}} \sum_{j=1}^{n_{\text{nodes}}} \phi_j(t) \right), \quad i = 1, 2, \ldots, n_{\text{nodes}} - 1, \quad (3.2)
\]
and $\theta_{n_{\text{nodes}}}(t) \overset{\text{def}}{=} -\sum_{j=1}^{n_{\text{nodes}}-1} \theta_j(t)$. (Hereafter, explicit time-dependence of $\theta(t)$ and $\phi(t)$ is usually not indicated, but can be assumed.) Since the vector $\hat{\omega}$ of natural frequencies is not time-dependent, the transformation is particular to this problem and not really relevant to the model reduction technique discussed in this chapter, though it ensures the existence of a steady $\theta$ state for sufficiently high values of $K$.

This transformation is used to generate a new dynamical system
\[
\frac{d\theta_i}{dt} = \dot{\omega}_i - \frac{1}{n_{\text{nodes}}} \left[ \sum_{j=1}^{n_{\text{nodes}}} \dot{\omega}_j \right] + \frac{K}{N} \left[ \sum_{j=1}^{n_{\text{nodes}}} A_{i,j} \sin(\theta_j - \theta_i) \right], \quad (3.3)
\]
\forall i \in [1, n_{\text{nodes}} - 1].

The main idea involves mathematical “technology transfer” from the field of UQ [41, 119], as discussed in §2.2. We assume that the long-time behavior of each unit in the assembly is characterized by (is a function of) its identity—the value(s) of the heterogeneous parameter(s). The problem can then be formulated as being distributed in (heterogeneous) parameter space ($\xi$-space) in a manner analogous to spatiotemporal processes “distributed” over physical space. The state $\theta_i(t)$ for any unit $i$ with identity $\xi_i$ can be approximated in terms of appropriate basis functions not in physical space, but rather in “identity” space: heterogeneous parameter space, as in Eqn. (2.6).

\[
\theta_i(t) = f(t; \xi_i) \approx \sum_{k=1, \ldots, \text{ncoeffs}} \alpha_k(t) \psi_k(\xi_i). \quad (3.4)
\]

The basis functions $\psi_k(\xi)$ are constructed as orthogonal polynomials in §3.3 and earlier in §2.2. Because we are modeling cases in which the behavior of each unit is
assumed to be a smooth function of identity (that is, nodes with similar identities are expected to behave similarly), a relatively short truncation of such a series may well be accurate enough if the right basis functions in $\xi$ space are chosen. In such a case, the number of ODEs to be solved reduces from the number of units $O(n_{\text{nodes}})$, to the number of terms in the series $O(n_{\text{coeffs}})$. This approach, and its links to UQ modeling/computational developments (like the use of Smolyak grids for nonintrusive collocation-based simulation) has been explored in [65]. Yet these developments were only applicable for all-to-all coupled, intrinsically heterogeneous assemblies of units.

The purpose of this chapter is to generalize this approach by introducing a simple, yet nontrivial, extension. Namely, we consider networks with non-trivial coupling structure, i.e. not all-to-all coupled. In this work all connections have the same strength; further extension to weighted connections is nontrivial, and the subject of current research. Each unit is now also characterized, beyond its ODE parameter values, by its connectivity in the network—the nature of its coupling with other units in the network, which in turn is quantified by features such as the unit’s degree (its count of undirected connections). Different nodes have different connectivity features (imposed by the network structure)—we can therefore think of connectivity as a type of heterogeneity of our building block units: structural heterogeneity rather than intrinsic heterogeneity.

It should be noted that the steady state of similar dynamical systems have been predicted analytically [45, 88]. However, [45] considers the $N \to \infty$ limit, whereas we deal with finite $N$. Additionally, [88] requires that all the phases be known exactly to use a self-consistency argument. We are trying to obtain a reduced description, so necessarily we do not know (nor want to know) all the phases of all oscillators, but rather to approximate them.

In the system Eqn. (3.1) to which we apply our coarse-graining strategy, we will see that the only connectivity feature that appreciably affects unit dynamics is the unit degree $\hat{\kappa}_i = \left( \sum_{j=1}^{n_{\text{nodes}}} A_{ij} \right) \in [0, N]$. That is, though we have previously shown that
intrinsically similar nodes in this system [71] (and others [70]) have similar dynamics with all-to-all coupling, we show here that, with a nontrivial coupling topology network, nodes which additionally have the same (similar) degrees also have similar dynamics (possibly after a short initial transient). The degree of a node can be treated as another heterogeneous node parameter, whose probability distribution is the network degree distribution. We argue that the same approach which uses the distribution of an intrinsic heterogeneity, and led to the reduction of all-to-all unit assemblies in [71] can be naturally extended to include a distribution over structural heterogeneity that leads to reduction of unit assemblies coupled in networks.

We demonstrate this in the simplest nontrivial representative setting we can put together: a set of coupled phase oscillators, characterized by heterogeneous frequencies $\hat{\omega}_i$ sampled from a prescribed distribution (here a truncated Gaussian)—but now not all-to-all coupled. Instead, the coupling $A$ is in the form of a complex network, generated by a Chung-Lu process similar to that described in [64, 17]. The ideas presented here work also for general networks, if the degrees of the nodes are large enough; the Chung-Lu network is used as a convenient example. Likewise, the particular degree distribution tested here (shown in Fig. 3.1) is not itself important, and networks with other degree distributions, such as power-law, or even nonmonotonic degree distributions, can also be used. Here, we first generate a weight sequence $w_i$, as

$$w_i = n_{\text{nodes}}p(1 - q(i - 1)/n_{\text{nodes}})^r, \quad i = 1, 2, \ldots, n_{\text{nodes}}$$  \hspace{1cm} (3.5)

with parameters $p = 0.50$, $q = 0.90$, and $r = 0.50$. With $w$, we generate the connection probabilities $P$ via

$$P_{ij} = P_{ji} = \min \left( \frac{w_i w_j}{\sum_k w_k}, 1 \right).$$  \hspace{1cm} (3.6)
We concretely generate an adjacency matrix $A$ from these probabilities by inverse transform sampling only above the diagonal of $A$, and copying to the bottom triangle, to ensure that the network is undirected, with no self-loops.

Instead of following the behavior of each individual oscillator, we exploit the observation that similar oscillators have similar behavior and can be tracked together. For all to all coupling, “similar oscillators” is taken to imply similar natural frequencies, and we write the oscillator state as a function of natural frequency [71] and time. However, when a non-trivially structured coupling exists, “similar oscillators” implies not only intrinsic similarity but also structural similarity. In addition to the intrinsic explanatory parameter of the natural frequencies, the degree of each node appears to be an explanatory parameter which suffices (in our simple model) to capture the influence of the coupling structure on the behavior of each oscillator; yet other features such as in-degree or local clustering coefficient are also worth considering. If two oscillators have similar $\hat{\kappa}$ values and similar $\hat{\omega}$ values, then their time-dependent behavior is observed here to be similar, possibly after a short transient. Finding the relationship between oscillator characteristics (intrinsic and structural) and oscillator states generates a coarse-grained description, whereby the system state can be encoded in fewer independent variables.

We illustrate a number of coarse-grained modeling tasks facilitated by this reduction: accelerated simulation (via coarse projective integration), accelerated fixed point computation, continuation, and coarse-grained stability analysis (via time-stepper based coarse Newton-Krylov GMRES [49, 50] and Arnoldi algorithms [93]). This creates a natural link between the reduction approach we present here, and our so-called Equation-Free framework for complex systems modeling [56, 55] (see §2.1).

In the end, what makes it all possible is the fundamental assumption about how heterogeneity (intrinsic as well as structural) affects the solution: “nearby” parameter values and “nearby” connectivities imply “nearby” dynamics. This is not always the
case for any network, and so testing that this assumption holds must be performed on a case-by-case basis. For our networks, such a check is demonstrated in Fig. 3.2. However, whether such a parameterization is possible is linked to the question of whether frequency-synchronization emerges, a subject for which an extensive literature exists [26].

The remainder of the chapter is organized as follows. First, we describe our illustrative example, a network of heterogeneous phase oscillators. We then give the form of our low-dimensional representation of the system state. We use our ability to transform back and forth between the two state representations to perform several computational tasks in the coarse-grained space, including the solution of initial value problems, the computation of fixed points, their stabilities and bifurcations. Appendices include an analysis of the validity of using higher-order coarse-grained integration schemes.

### 3.2 An illustrative example of heterogeneous coupled oscillator networks

Our illustrative example is a network of coupled Kuramoto oscillators with heterogeneous natural frequencies $\hat{\omega}_i$, coupled in a stochastically generated network (here, a Chung-Lu network [17] with parameters $p = 0.50$, $q = 0.90$, and $r = 0.50$, an example instance of which is shown in Fig. 3.1). This type of model system was used in some previous reduction studies [71, 86]. The number of oscillators in the network is also a parameter we will vary; our base case is $n_{\text{nodes}} = 196$. A basic premise, which is corroborated by Fig. 3.6(a), is that the network is large enough (the number $N$ of nodes is large enough) for the single realization to be representative of the expectation over all consistent network realizations. The fine dynamics are governed by the system of coupled ordinary differential equations (ODEs) Eqn. (3.1),
where the natural frequencies $\hat{\omega}_i$ and the node degrees (numbers of neighbors) $\hat{\kappa}_i$ are heterogeneous across the oscillators constituting the network. We remind the reader of our assumption that, of all structural node features that may affect the dynamics, it will be the node degree that matters here—so that the identity of node $i$ is sufficiently described by its intrinsic parameter $\hat{\omega}_i$ and its structural parameter $\hat{\kappa}_i$. This assumption is supported first by Fig. 3.2, and later, as we will see, more quantitatively by Fig. 3.6(a).

Figure 3.1: **Visualization of a Chung-Lu network** [17] with $n_{\text{nodes}} = 4000$ nodes, constructed using parameters $p = 0.50$, $q = 0.90$, and $r = 0.50$. In 3.1(a), the network is plotted with MATLAB’s 2D spectral projection-based layout. In 3.1(b), the degree histogram is shown.

We further define rescaled versions of the two heterogeneous parameters, $x_i = (\hat{x}_i - \text{mean}(\{\hat{x}_j\}))/\text{stddev}(\{\hat{x}_j\})$ for $x = \omega, \kappa$ and $i, j \in 1, \ldots, n_{\text{nodes}} - 1$. These two transformations do not affect the fine dynamics of Eqn. (3.1) or Eqn. (3.3); only the numerics of the implementation of the restriction $R$ to a coarse-grained state representation, to be developed below. Without axis markings, Fig. 3.2, for instance, would look the same whether $\omega \times \kappa$ or $\hat{\omega} \times \hat{\kappa}$ were used for plotting.

The emergent functional dependence of the $\theta_i$ on the $\omega_i$ (the intrinsic heterogeneity only) was discussed in the all-to-all coupling context in [72, 71]. There, we used a one-dimensional polynomial chaos expansion (PCE) to describe the reduced problem
for $A_{ij} = 1 \forall i, j \in [1, n_{\text{nodes}}], i \neq j$, so that all nodes have degree $\kappa_i = n_{\text{nodes}} - 1$. In this chapter, we again expect the oscillator states to quickly become smooth time-dependent functions of their identities, but node $i$’s identity now includes both $\omega_i$ and $\kappa_i$. For $K$ sufficiently large so that a steady state of Eqn. (3.1) exists, as in Fig. 2.1, we indeed observe that the states of randomly initialized oscillators quickly approach an apparently smooth surface in $\omega \times \kappa$ space (see Fig. 3.2) suggesting that a low-order series truncation of the type described in Eqn. (3.8) may constitute a good description. This motivates the use of a functional fit of the coefficients (the few $\alpha_k(t)$ in Eqn. (3.8)) to the data (the many $\theta_i(t)$) as a coarse representation.

In previous work [86] we have used a projection onto the eigenvectors of the discrete Laplacian on the graph to describe the dependence of oscillator state on structural heterogeneity, while using a one-term/linear fit to account for dependence on intrinsic frequency.

Figure 3.2: Oscillator states (phases) quickly slave to oscillator “identities”. The network oscillator states are initialized as a cloud in $(\hat{\omega}, \hat{\kappa}, \theta(t))$ space, and are clearly seen to quickly rearrange onto a 2-D surface. Points are colored by $\theta$ value. $n_{\text{nodes}} = 4000$ and $K = 0.5$ were used. Inset plots show slices at high and low $\hat{\kappa}$ values, including all oscillators in two bands of width 100.
3.3 Low-dimensional representation

Polynomial chaos

Given our observation that oscillator behavior quickly becomes a function of oscillator identity, we want to describe the long-term dynamics of the oscillator phase angles as a smooth function \( \theta = \theta(t; \omega, \kappa) \). The phase angle of the \( i \)-th oscillator is then given by \( \theta_i(t) \equiv \theta(t; \omega_i, \kappa_i) \). Since our two heterogeneities (the intrinsic and the structural) are here independent, the basis functions are a tensor product of two independent polynomial bases

\[
\psi_{\gamma}(\omega, \kappa) = \xi_{\gamma_k,\omega}(\omega)\zeta_{\gamma_k,\kappa}(\kappa).
\]

The orthogonality of this basis devolves on the orthogonality of the individual one-dimensional bases. This is a special case; the formulation will still in principle be applicable for parameters with correlated joint probability distributions if one constructs an appropriate set of basis functions [77, 24]. Both of these cases are discussed in detail in §2.2.

We now express the network dynamics in the form of a series expansion in a truncation of this tensor product basis as

\[
\theta(t) \approx \sum_{k=1}^{n_{\text{coeffs}}} a_k(t) \psi_k(\omega, \kappa) \equiv \sum_{k=1}^{n_{\text{coeffs}}} a_k(t) \xi_{\gamma_k,\omega}(\omega)\zeta_{\gamma_k,\kappa}(\kappa) \\
G = \{ \gamma_k = (\gamma_k,\omega, \gamma_k,\kappa) \} : 0 \leq \gamma_k,\omega, \gamma_k,\kappa \in \mathbb{Z}, \gamma_k,\omega + \gamma_k,\kappa \leq p_{\text{max}} \}, \quad (3.8)
\]

\[
n_{\text{coeffs}} = ||G|| = (1 + p_{\text{max}})(2 + p_{\text{max}})/2
\]

where the \( a_k(t) \) are time-dependent coefficients, \( \xi_{\gamma_k,\omega}(\omega) \) are basis functions arising from the intrinsic heterogeneity dependence and \( \zeta_{\gamma_k,\kappa}(\kappa) \) are basis functions arising from the structural heterogeneity dependence. Within the truncation of the set of functions \( G \) included in the basis, the ordering of the basis can be chosen arbitrarily, and so we substitute the vector index \( \gamma_k = (\gamma_k,\omega, \gamma_k,\kappa) \) with a scalar index \( 1 \leq k \leq n_{\text{coeffs}} \).
The analogy with UQ now manifests itself in our choice of the two independent basis sets: each one of them is chosen to be a polynomial chaos basis in the corresponding heterogeneous (in analogy to random) parameter. Each set of polynomials is orthogonal with respect to the probability density of the corresponding heterogeneous parameter, and the joint heterogeneity probability density is just the product of the two unidimensional, independent heterogeneity probability densities, so that Eqn. (2.8) is satisfied, as shown in §2.2.

Note that in Eqn. (3.8), we specify that $\gamma_{k,\omega} + \gamma_{k,\kappa} \leq p_{\text{max}}$. This allows us to say that our two-dimensional polynomials are of total degree $\leq p_{\text{max}}$. An alternative truncation rule would be to require that $\gamma_{k,\omega} \leq p_{\text{max}}$ and $\gamma_{k,\kappa} \leq p_{\text{max}}$ (or even to place separate bounds on $\gamma_{k,\omega}$ and $\gamma_{k,\kappa}$, allowing for some anisotropy in the details, a topic for separate investigation). Both approaches can be found in the literature.

This allows, per Eqn. (3.8), an approximation of the behavior as a time-dependent two-dimensional surface in one intrinsic dimension (here, the (normalized) natural frequencies $\omega$), and one structural dimension (here, the (normalized) node degrees $\kappa$). We repeat (see §2.2) that this tensor product basis, limited to those polynomials of total order less than some desired maximum, is a truncated orthogonal basis for the 2D space weighted by probability densities that are products of two marginal distributions.

In classical Galerkin methods an inner product is taken between the governing evolution equations and each basis function, producing ODEs for the dynamics of the expansion coefficients (the $\alpha_k(t)$ in Eqn. (3.8)) by exploiting orthogonality. A similar approach could be taken here (through analytical computation of the inner product integrals if possible, else through numerical quadrature). This approach is examined in more detail in chapter 4. Instead, we do not directly calculate the temporal rates-of-change of the expansion coefficients, but infer features of the coefficient dynamics from brief bursts of simulation of the dynamics of the full system (see §3.4). This equation-
free approach relies on our ability to go back and forth between fine descriptions of the system state (the $\theta_i$ values), and coarse ones (the $\alpha_k$ values). This is analogous to a non-intrusive (black-box, input-output) approach to using polynomial chaos in UQ.

**Equation-free numerics**

The choice of the polynomial basis sets follows the selection of the appropriate heterogeneity distribution. For several frequently encountered distributions, the bases have been tabulated (e.g. Table 2.1) from original generalized polynomial chaos references (see e.g. [119]). If not already available in such tables, one can construct the basis polynomials e.g. through Gram-Schmidt orthogonalization with inner products in the space weighted by the distribution function. A couple of nontrivial considerations arising in our case are that (a) our structural heterogeneity parameter (the node degree) takes integer values, and so the degree distribution is discrete; and (b) often we may encounter problems for which the heterogeneity distribution is not explicitly known, but has to be estimated from specific system realizations (so, from large enough samples). For the case of explicitly unknown but sampled distributions (whether discrete or continuous) we have used here the moments of the sampling of the heterogeneity parameters for our particular network realization to extract the corresponding polynomials (using SVD-based pseudo-inverses) [79].

Results of the moment-based polynomial generation method, which we used to generate all 1D polynomials used in this chapter, are shown in Fig. 3.3. Here, the marginal samplings of degrees and natural frequencies were used separately to generate 1D polynomials via moments, and then a 2D basis was defined from the tensor product of these 1D bases, with the restriction $O(\omega) + O(\kappa) \leq p_{\text{max}}$ placed on the total polynomial degree of the 2D basis functions used.

For the expansion in Eqn. (3.8), we used a set of 2D basis functions. Once the relevant polynomials have been constructed, the appropriately defined inner
product also allows us (whether for continuous or for discrete distributions, explicitly known or not) to find the coefficients $\alpha_k$ in Eqn. (3.8) for a given observation $\theta(t)$ of the system states for a particular system realization (sampling of the distribution). This can be accomplished directly (via numerical approximations of the relevant inner products), using the orthogonality of the $\psi_k$, as described in Eqn. (2.7). For our problem, with one continuous and one discrete variable, this can be written concretely as

$$\int_{\Omega} g(x) d\Gamma(\xi) = \int \sum_{\kappa=0}^{n_{\text{nodes}}} g(\kappa, \omega) \rho_\kappa(\kappa) \rho_\omega(\omega) d\omega, \quad (\omega, \kappa) \in D,$$

where $\rho_\kappa$ is the (discrete) probability mass function for the degrees, and $\rho_\omega$ is the (continuous) probability distribution function for the natural frequencies. In chapter 4, we examine the computation of this integral for the case when the problem can be recast as a PDE, and so the coupling sum in Eqn. (3.1) and Eqn. (3.3) can also be written as an Lebesgue integral. There, we consider standard Monte Carlo integration in addition to Gaussian quadrature and a repurposing of anchored ANOVA. These latter methods have the benefit of allowing integrals to be computed using only a few key virtual oscillators, with anchored ANOVA having the additional benefit of decreased scaling sensitivity to the number of random dimensions (2 in this chapter). However, these
benefits require that the original model be recast as PDEs continuous in both time and the random dimensions.

Here, we take the alternate approach of finding the $\alpha_k$ indirectly through least squares fitting, minimizing the squared residual norm $\sigma$ with respect to the coefficients $\alpha_k$ (here using a QR algorithm).

\[ \sigma = \left\| f(x) - \sum_{k=1}^{n_{\text{coeffs}}} a_k \psi_k(x) \right\|^2_2 \]

\[ \approx \hat{\sigma} = \sum_{i=1}^{n_{\text{samp}}} w(x_i) (f(x_i) - \sum_{k=1}^{n_{\text{coeffs}}} \alpha_k \psi_k(x_i))^2, \] (3.9)

where $\lim_{n_{\text{samp}} \to \infty} \hat{\sigma} = \sigma$ according to the law of large numbers, and the weights $w(x_i)$ are still to be decided.

For a (large) finite sample $x_i, i = 1, \ldots, n_{\text{samp}}$, if we take the partial derivative $\partial \hat{\sigma} / \partial \alpha_k$, and use the fact that $\sum_{i=1}^{n_{\text{samp}}} w(x_i) \psi_l(x_i) \psi_k(x_i) = 0$ if $l \neq k$ (orthogonality) to remove some terms, then we find that

\[ \alpha_k = \frac{\sum_{i=1}^{n_{\text{samp}}} w(x_i) f(x_i) \psi_k(x_i)}{\sqrt{\sum_{i=1}^{n_{\text{samp}}} w(x_i) \psi_k(x_i)^2}}, \] (3.10)

So, as long as we accept that

\[ \sum_{i=1}^{n_{\text{ samp}}} w(x_i) g(x_i) \] (3.11)

is a good approximation to

\[ \int_{\Omega} g(x) d\Gamma(\xi), \] (3.12)

we obtain the same formulas for the $\alpha_k$.

Suppose $x$ has a density $\rho(x \in D)$, so Eqn. (3.12) can be written as

\[ \int_{\Omega} g(x) \rho(x) dx. \] (3.13)

If the $x_i$ are chosen randomly in accordance with $\rho(x)$, then Eqn. (3.11), where $w(x_i) = 1/n_{\text{ samp}}$, is a good approximation to Eqn. (3.12). This is just Monte Carlo
integration, and the law of large numbers gives

$$\lim_{n_{\text{samp}} \to \infty} \frac{1}{n_{\text{samp}}} \sum_{i=1}^{n_{\text{samp}}} g(x_i) = \int_{\Omega} g(x) \rho(x) \, dx.$$ (3.14)

### 3.4 Coarse computational modeling tasks

As discussed in more generality in §2.1, here we employ a coarse-graining scheme to summarize system states. We define restriction and lifting operators $R$ and $L$ as in §2.1. Here, $R$ operates by minimizing the residual $\sigma(\alpha(t))$ from Eqn. (3.9), and $L$ maps $\alpha$ vectors to $\theta$ vectors by setting the $\theta_i$ values equal to the right-hand-side of the approximant in Eqn. (3.8) evaluated at the corresponding $(\omega_i, \kappa_i)$. The $L$ and $R$ operators combine to define a coarse timestepper $\Phi_{\tau,C}$, as defined in Eqn. (2.1).

**Coarse initial value problems**

Given the coarse timestepper $\Phi_{\tau,C}$ of Eqn. (2.1), we can accelerate the computation of dynamic trajectories of the system, through CPI as described in §2.1 (for example, the coarse Euler method of Eqn. (2.2)). It is important to note here that in all our CPI computations we used a single network (with a single $\omega$ vector) to generate the polynomial basis functions and to lift to at every projective step. This can be thought of as a “single instance” CPI; one may also consider CPI for the expected behavior over all networks that share the same degree as well as $\omega$ distributions – in which case one should lift to many consistent network realizations and average over them. This issue will be examined more closely below. Results of applying the simpler scheme to the coupled oscillator problem with coarse variables obtained by a 2D PCE fit are shown in Fig. 3.4.

This approximation of the coarse right-hand-side function can be used for other computational tasks besides the computation of dynamic trajectories.
Figure 3.4: **Coarse projective integration** shows smooth evolution of the first few leading PCE coefficients \( \alpha_j \), with some corresponding fine states visible in the 3D insets. Black curves (in the main figure) and dense grey scatters (in the insets) were obtained by full fine integration with the same initial \( \theta \) conditions using MATLAB’s \texttt{ode23}. Colored points (in the main figure) and red surfaces (in the insets) were obtained via CPI, using several different integrators. At each coarse step, \( \frac{d\alpha_k}{dt} \) was estimated \( \forall k = 1, \ldots, n_{\text{coeffs}} \) (where \( n_{\text{coeffs}} = 28 \)) by drawing \( n_{\text{coeffs}} \) chords through the restrictions of the last two points in a brief burst of fine integration of \( \tau = 0.05 \) time units. At the times indicated, we make inset plots with red surfaces corresponding to the lifted CPI state and grey scatters corresponding to the closest (in time) state in the true trajectory. These should be compared to Fig. 3.2. We performed the same task for several outer integrators: two explicit Runge-Kutta integration schemes, and a coarse wrapper around the built-in MATLAB integrator \texttt{ode45} are compared to the restrictions of points in the fine trajectory starting from the same lifted initial condition. For the two explicit Runge-Kutta integrators, an outer step of \( h = 0.45 \) was used. For \texttt{ode45}, an absolute tolerance of \( 10^{-6.0} \) and a relative tolerance of \( 10^{-12.0} \) were used. \( n_{\text{nodes}} = 300, K = 1, \) and \( n_{\text{coeffs}} = 28 \) were used. The \( \omega \) values were drawn from a truncated normal distribution supported on \([-0.100, 0.100]\), with zero mean and standard deviation 0.060.

**Coarse fixed point computation**

The coarse time-stepper can be used to define a coarse difference as given in §2.1 and Eqn. (2.3). Zeros of the coarse difference map correspond to coarse steady states of the original problem.

A Newton-Krylov iteration to find such a state is depicted in Fig. 3.6. In Newton-Krylov GMRES (Generalized Minimal RESidual), the inner linear problem of an outer (nonlinear) Newton-type solver is solved by GMRES, in which the solution to \( Bx = b \)
Figure 3.5: As the inner step size $\tau$ decreases, the error (compared to direct integration) of a projective (not coarse-projective) integration becomes bounded by the integrator’s intrinsic (outer) step size $h$. The true solution at $t = 0.417$ was found by integrating using MATLAB’s ode45 with an absolute tolerance of $10^{-12.0}$ and a relative tolerance of $10^{-12.0}$. The series of black circles give the error at $t = 0.417$ that results from using integration using the true RHS function Eqn. (3.3) in an explicit second-order Runge-Kutta integration scheme of (outer) step size $h$. The colored curves use the same integrator and outer step size, but approximate the RHS function with the difference map $f_\tau(\theta(t)) = \theta(t) - \Phi_{\tau,F}[\theta(t)]$, analogous to the coarse difference map of Eqn. (2.3). Error was evaluated by taking the norm of the vector difference between the projective integration solution $\theta(t = 0.417)$ and the true solution.

is assembled in a space derived from the $n^{th}$ Krylov subspace $\{B^j r_0\}_{j=0,\ldots,n-1}$, where $r_0$ is the residual of the initial iterate (and $B$, which is not computed, is the Jacobian of Eqn. (2.3)).

Here we work again with $n_{\text{nodes}} = 196$ node networks. The basis polynomials are computed from a single network realization (a large, 10000 node Chung-Lu network); because the support of the degree distributions for a $n_{\text{nodes}} = 10000$ and for a $n_{\text{nodes}} = 196$ network are not the same, the degrees and frequencies are normalized as described in §3.2. But now we construct 32 realizations of networks consistent with the chosen degree distribution, and perform our fixed point computation for each one of them. We do not regenerate polynomials for each of these realizations; we observe them on the “large sample” polynomials; this is justified in Fig. 3.6(b), where we recover the Probabilist’s Hermite polynomials as $n_{\text{nodes}}$ increases, by generating with sample moments. The effect of regenerating polynomials within a larger computation is considered further in [87, 85].
We report the sensitivity of the results to the basis size $n_{\text{coeffs}}$ in Fig. 3.6(a), where the error bars are indicative of the variation across our 32 network samples. We can also see in Fig. 3.6(a) that the dimension of the Krylov subspace (at the final Newton iteration before convergence) initially closely follows the size of the 2D basis we use (it uses "all of $n_{\text{coeffs}}$") but later on plateaus.

![Figure 3.6](image)

**Figure 3.6:** 3.6(a) Coarse fixed point computations, and their sensitivity to the size of the coarse basis used. Computations were repeated for different samples of network and natural frequencies and the error bars are indicative of the resulting variation in the solution. Error was computed between lifted fixed points of Eqn. (2.3) (with $\tau = 0.05$), and steady states of Eqn. (3.3). The norm used was the mean squared error (MSE) across the $N - 1$ nodes. 32 replicates were used per value of the independent variable, an absolute tolerance of $10^{-6.0}$ was used for the outer Newton solver, error bars are $\pm$ 1 standard deviation, and $K = 1$ was used. 3.6(b) Convergence of the orthogonal polynomials based on (increasingly larger) finite networks. As $n_{\text{nodes}}$ increased, the polynomials generated via sample moments approached the $n_{\text{nodes}} = 10000$ polynomials. Error was quantified in the 2-norm of the monomial coefficient tensor $C$ in $\psi_k(\omega, \kappa) = \left(\sum_{l=0}^{p_k,\omega} C_{\omega,k,l} x^l\right) \left(\sum_{l=0}^{p_k,\kappa} C_{\kappa,k,l} x^l\right)$, where the pair $p_k = (p_{k,\omega}, p_{k,\kappa})$ gives the orders of the two one-dimensional polynomials.

**Coarse stability computations: eigenvalues and eigenvectors**

(Coarse) eigenvalues of the Jacobian of the (coarse) difference map Eqn. (2.3) upon convergence to its (coarse) fixed points can be used to establish the stability of said fixed points and help determine the nature of their potential (coarse) bifurcations.
These eigenvalues $\mu_i$ are related by

$$\lambda_i \approx \hat{\lambda}_i = \ln(\mu_i + 1)/\tau$$

(3.15)

to the corresponding eigenvalues $\hat{\lambda}_i$ of the Jacobian of the (unavailable) coarse differential evolution equations; in turn, these should coincide with the leading eigenvalues $\lambda_i$ of the actual problem (the leading eigenvalues of the fine differential equations).

For $n_{\text{coeffs}} > 3$ coarse eigenvalues $\mu_i$ were obtained through the Jacobian-free implicitly restarted Arnoldi Method (IRAM) [49, 67] applied to the coarse difference operator Eqn. (2.3). For $n_{\text{coeffs}} \leq 3$, a forward finite-difference Jacobian with a fixed step size of 0.001 was computed and the eigenpairs calculated directly with the QZ algorithm implemented in MATLAB’s \texttt{eig}.

As the number of coarse variables is increased, and therefore the quality of the coarse approximation improves, one expects these coarse eigenvalue estimates to approach the leading eigenvalues of the analytical fine Jacobian of Eqn. (3.3), located through any eigensolver. In Fig. 3.7(a), we demonstrate this convergence of the approximate eigenvalues $\hat{\lambda}_i$ to the leading fine eigenvalues $\lambda_i$ with increasing $n_{\text{coeffs}}$. Fig. 3.7(b) shows the corresponding convergence of a coarse eigenvector (the one corresponding to the smallest absolute value of $\hat{\lambda}_i$) to the fine eigenvector corresponding to smallest absolute value of $\lambda_i$. Note that lifting is necessary to make a comparison between $\theta$ and $\alpha$ eigenvectors. On the other hand, the transformation Eqn. (3.15) is necessary not because of our coarse and fine spaces, but because the $\mu$ eigenvalues come from the Jacobian of a finite-time flow map while the $\lambda$ eigenvalues come from the Jacobian of a vector of infinitesimal differential equations.

In performing computations involving finite differences, we used a value of $\sqrt{\text{machine precision}}$ (according to [84]), which is approximately $10^{-7}$ for IEEE standard double-precision floating point variables in MATLAB.
Figure 3.7: Comparison of coarse and fine eigencomputations for different basis sizes close to (left column) and far away from (right column) the main SNIPER bifurcation (see text). Eigenpairs obtained from Eqn. (2.3) (with $\tau = 0.05$, $n_{\text{nodes}} = 196$, and $n_{\text{coeffs}} = 28$) are similar to those obtained from Eqn. (3.3). True eigenvalues and eigenfunctions (small blue points) were obtained using MATLAB’s eig on an analytical Jacobian of Eqn. (3.3). Approximate eigenvalues (large red diamonds) and eigenvectors (smooth surfaces) were obtained via implicitly restarted Arnoldi iteration (IRAM), with the transformation Eqn. (3.15). Converge to the the fine eigenvalues $\lambda_i$ as $M$, the number of $\alpha_k$ coefficients, rises. For larger values of $K$, the eigenvalues are all increasingly negative, though the ratio between the first and second eigenvalue (about 0.6) does not change by much. In 3.7(a), the horizontal axis of each plot is an index across eigenvalues, while the vertical axis is eigenvalue. In 3.7(b), the lifted view of the leading coarse eigenvector visually approaches the leading fine eigenvector. The coarse eigenvector was evaluated as a surface (in a manner similar to Eqn. (3.8)) at a fine grid of points within the convex hull of the sampled $(\omega, \kappa)$ points. The “eigensurface” corresponding to the slowest eigenvalue appears to approach an indicator function on the oscillator whose extreme $(\omega, \kappa)$ pair makes it the most susceptible to “desynchronization” with decreasing $K$. Eigenpairs were chosen to match the right (synchronized) inset plot in Fig. 3.8, and the point closest to the turning point along the branch of coarse fixed points in that figure.

Coarse continuation and bifurcation diagrams

To build a coarse bifurcation diagram (see Fig. 3.8), we performed pseudo-arclength continuation [48, 51] for the coarse fixed points. We computed branches of coarse solutions to $0 = F_{\tau}(\alpha; K)$ as the global parameter $K$ is varied. To trace out these solution branches, steps were taken in (pseudo-)arclength along the branch rather than in $K$. This allows the continuation to extend naturally beyond turning points. At
some point along this continued branch of solutions, one of the computed eigenvalues becomes positive. At this point, the line color is changed to indicate that the new branch comprises unstable solutions.

Figure 3.8: **Bifurcation diagram, coarse and fine.** Coarse fixed-point solutions can be used to generate a bifurcation diagram in the parameter $K$, via pseudo-arclength continuation of fixed points of the coarse flow map in Eqn. (2.1) (with $\tau = 0.30$, $n_{\text{nodes}} = 196$, and $n_{\text{coeffs}} = 28$). At the point where the curve changes from solid to dashed, one eigenvalue passes through zero. This marks a change from a stable to an unstable branch. The two inset plots show representative trajectories of the real magnitude $r$ of the complex synchronization index $r e^{i\theta}$.

Beyond these branches (to the left) we know that a limit cycle solution arises: a periodic orbit characterized by one free rogue oscillator, which performs full rotations and only momentarily slows down as it passes through the remaining pack of clustered oscillators [72]. In dynamical systems terminology this is a “SNIPER” (saddle-node infinite period) bifurcation [104]. The insets in Fig. 3.8 show transient dynamics in terms of the synchronization index $S$ in $S e^{i\theta} = \frac{1}{n_{\text{nodes}}} \sum_{j=1}^{n_{\text{nodes}}} e^{i\theta_j}$ (the real magnitude of the complex Kuramoto order parameter; see e.g. [102]). The presence of this
rogue oscillator means that the coarse representation of Eqn. (3.8) is not particularly accurate/informative to the left of $K_c$, without explicitly including the rogue’s value of $\theta$ in the set of coarse variables, as was done, for example, in [71].

3.5 Discussion

In this chapter we have demonstrated that a general network of coupled, intrinsically heterogeneous, oscillators can be usefully described using a small number of collective dynamic variables. These variables are the time-dependent coefficients of an expansion of the complete state of the network in terms of a set of orthogonal polynomials. The polynomials are products of univariate polynomials in the parameters describing the intrinsic heterogeneity of a given oscillator, and a structural heterogeneous property (here, the degree) indicative of the connectivity of the oscillators in the network. Our results extend previous work which only considered all-to-all coupled networks, in which the state of an oscillator was a function of only its intrinsic heterogeneity [71, 63]. Our expansion (and subsequent truncation of the expansion) in this form is motivated by the large body of work in the field of uncertainty quantification; the difference being that here we have heterogeneous parameters characterizing a single network, rather than many realizations of a dynamical system, each with different (uncertain) parameters. We anticipate that this new link between the two fields (network dynamics and UQ) may provide many more fruitful opportunities for mathematical/computational technology transfer that can enhance our understanding and ability to usefully describe and analyze dynamics on complex networks.

Although we have only considered Kuramoto-type oscillators in a specific Chung-Lu network, our methods do not rely on either the type of oscillator used or the specific network (as long as the mean degree is not small). Thus they should be widely
applicable to many non-trivial networks of neurons which exhibit synchrony for some range of parameters.

Using this reduced description of a network, we demonstrated a number of standard computational tasks using the equation-free framework, in which differential equations describing the evolution of the expansion coefficients are not explicitly derived, but rather estimated on-the-fly. Specifically, we demonstrated coarse projective integration, the computation of coarse fixed points and their stability, as well as parametric analysis through continuation.

The success of our method relied on the rapid development of correlations between the state of an oscillator and its heterogeneous identifying parameters, in this case, its intrinsic frequency and its degree. A potential shortcoming of the method would arise when such a strong dependence does not develop—that is, when “similar” oscillators do not behave “similarly” (e.g. when the initial conditions, or something more than just the degree, like the clustering coefficient of every node, matters). This implies that additional “heterogeneity dimensions” must be introduced, in analogy to when, say, a two-dimensional flow loses stability and becomes three-dimensional. One such case we have encountered [70] is when the oscillators in a network (an all-to-all network of Hodgkin-Huxley neurons) split in two subsets and in each subset a distinct relation of state to identity was established. Knowing the identity of the oscillator was not enough, in that case, to characterize dynamics; one also needed to know in which cluster the oscillator belonged. We could regard both our case as well as this other one as special cases where, for every oscillator identity, there is a distribution of behaviors—a strongly peaked unimodal distribution in this chapter, and a strongly peaked bimodal distribution in [70]; for that matter, in our study of breakup in multiple communities/clusters, one obtained a multimodal distribution. We anticipate that, in the spirit of stochastic PDEs in physical space, our approach might be extended.
to evolve state distributions in heterogeneity space (as opposed to state functions in heterogeneity space).

In other networks it may be that the state of a node depends on more than just these two properties: networks with weighted edges provide an obvious context in which this may occur. We believe (and are actively pursuing this research direction) that the approach introduced here can also be usefully extended to help in determining reduced descriptions for such networks.

The tensor product basis used here relied on a lack of correlation across the heterogeneities. As mentioned in §3.3, this reliance can be overcome by generating the full multidimensional basis all at once, and our current work addresses this possibility. This is likely to be the case in dynamical systems in which it is useful to retain multiple structural heterogeneities. Degree is one of several structural parameters–others include a node’s participation in motifs like triangles (complete graphs on three nodes), cherries (triangles with one edge removed) or its local clustering coefficient. This progression can be continued to higher-order statistics of the node connectivity by noting that using the degree of each node as the representative structural heterogeneity is equivalent to considering the per node counts of the two-node one-edge motif. As more structural heterogeneities are considered, it is reasonable to expect that these heterogeneities will not be statistically independent.
Chapter 4

Analysis of variance for intrusive polynomial chaos expansion

4.1 Intrusive PCE for dynamics

Minimal PDE setting

Consider a PDE

\[
\frac{\partial}{\partial t} \theta(\xi; t) = \begin{bmatrix}
\frac{\partial}{\partial t} \theta^{(1)}(\xi^{(1)}, \ldots, \xi^{(d_u)}; t) \\
\vdots \\
\frac{\partial}{\partial t} \theta^{(nvars)}(\xi^{(1)}, \ldots, \xi^{(d_u)}; t)
\end{bmatrix} = f(\theta(\xi, t), \xi, t)
\]

(4.1)

(recall Eqn. (2.5)), where the several functions \(\theta^{(i)}\) represent quantities such as temperature, pressure, or electric potential; these functions are evaluated over a space \(\Omega \ni \xi\) (physical or configurational and weighted by a density \(\rho(\xi) = d\Gamma(\xi)/d\xi\)); and where \(f\) generally includes integrals or partial derivatives of \(\theta^{(k)}\) over or with respect

\*This work was published in [16].\*
to $\xi(t)$. Treated as an initial value problem, one way to solve Eqn. (4.1) is to discretize into a finite set of grid points $\xi_i$, after which the PDE is approximated by a collection of ODEs coupled across space.

For problems such as that discussed in §4.3, in which $f$ contains an integral over $\Omega$, quadrature theory [84] can guide the placement of these grid points. With a particular weighting function $\rho$, Gaussian quadrature can supply both the grid points $\xi_i$ and a set of weights such that integrals over the space can be approximated with an explicit, weighted sum. (A simple case of such an explicit sum approximating a integral is the trapezoidal rule.) However, the number of grid points required from Gaussian quadrature to compute integrals to a required error can be large, and particularly this number scales poorly with the dimension $d_u$ of the space. Another approach is to approximate these integrals via analysis of variance (ANOVA) is discussed in §4.2.

An alternative is to use a basis expansion, rather than finite-difference discretization to represent functions of $\xi$. Here, functions are represented by a truncated sum over a carefully chosen basis:

$$\theta^{(k)}(\xi) \approx \sum_{j=1}^{n_{\text{coeffs}}} \alpha_j^{(k)} \psi_j(\xi). \quad (4.2)$$

Because, for some problems, they are often the eigenfunctions of the dynamics, sines and cosines are sometimes used in such a scheme. However, for our purposes (where the weighting function $\rho$ is a probability distribution), other bases are more apt.

**ODEs for expansion coefficients $\dot{\alpha}_k(t)$**

Embedding the expansion of Eqn. (4.2) into Eqn. (4.1), we can obtain ODEs for the expansion coefficients (as in Eqn. (2.9)) Recalling Eqn. (2.7), where we showed that the fitting coefficients are given by the inner product between the fitted function and
the basis functions, we now perform a similar projection with the \( \theta \) ODE.\(^1\)

\[
\frac{\partial}{\partial t} \theta(\xi; t) \approx \frac{\partial}{\partial t} \left[ \sum_{j=1}^{n_{\text{coeffs}}} \alpha_j(t) \psi_j(\xi) \right] 
= \sum_{j=1}^{n_{\text{coeffs}}} \frac{d\alpha_j(t)}{dt} \psi_j(\xi) 
\] (4.3a)

\[
\langle \frac{\partial \theta}{\partial t}, \psi_k \rangle \rho \approx \int_{\Omega} \left( \sum_{j=1}^{n_{\text{coeffs}}} \frac{d\alpha_j(t)}{dt} \psi_j(\xi) \right) \psi_k(\xi) d\Gamma(\xi) 
= \int_{\Omega} \sum_{j=1}^{n_{\text{coeffs}}} \left( \frac{d\alpha_j(t)}{dt} \psi_j(\xi) \psi_k(\xi) \right) d\Gamma(\xi) 
= \sum_{j=1}^{n_{\text{coeffs}}} \left( \frac{d\alpha_j(t)}{dt} \int_{\Omega} \psi_j(\xi) \psi_k(\xi) d\Gamma(\xi) \right) 
= \sum_{j=1}^{n_{\text{coeffs}}} \left( \frac{d\alpha_j(t)}{dt} \langle \psi_j(\xi), \psi_k(\xi) \rangle_{\rho} \right) 
= \sum_{j=1}^{n_{\text{coeffs}}} \left( \frac{d\alpha_j(t)}{dt} \delta_{jk} \right) 
= \frac{d\alpha_k(t)}{dt} 
\] (4.3b)

So, if the integral of the inner product can be computed analytically, or numerically to a required tolerance, we can evaluate ODEs for the coarse coefficients. These ODEs can then be integrated themselves (most likely with a numerical timestepper).

### 4.2 ANOVA for computing inner products

ANOVA is widely used as a statistical method to test differences between two or more means [44, 32]. The same idea can be used for the interpolation and integration of high dimensional problems as well as analyzing stochastic simulations. [34, 103]. In §2.2, we showed that ODEs for the coefficients of an approximating sum could be evaluated

\(^1\) Note that, as in §2.2, bracketed superscripts are excluded from \( \theta \) and \( \alpha \) in Eqn. (4.3). However, for uses like the pre-Bötzinger model of Eqn. (4.14) and Eqn. (4.21) where \( n_{\text{vars}} > 1 \), this analysis is repeated \( n_{\text{vars}} \) times.
with high precision if a particular inner product integral could be computed. The usual method for computing this integral in stochastic spaces of moderate dimension (below about $d_u = 10$) is Gaussian quadrature. However, for as the dimension of the heterogeneity rises, this becomes infeasible. Instead, we repurpose ANOVA to approximate this integral as a sum of evaluations of lower-dimensional functions.

Consider an integrable function $\theta(\xi), \xi = (\xi_1, \xi_2, \cdots, \xi_{d_u})$ defined in $I_{d_u} = [0, 1]^{d_u}$. The ANOVA representation for $\theta(\xi)$ is as follows:

**Definition 4.2.1** The representation of $\theta(\xi)$ in a form

\[
\theta(\xi) = \theta_0 + \sum_{s=1}^{d_u} \sum_{j_1 < \cdots < j_s} \theta_{j_1 \cdots j_s}(\xi_{j_1}, \cdots, \xi_{j_s}) \tag{4.4}
\]

or equivalently

\[
\theta(\xi) = \theta_0 + \sum_{1 \leq j_1 \leq d_u} \theta_{j_1}(\xi_{j_1}) + \sum_{1 \leq j_1 < j_2 \leq d_u} \theta_{j_1,j_2}(\xi_{j_1}, \xi_{j_2}) + \cdots + \theta_{1,2,\ldots,d_u}(\xi_1, \xi_2, \cdots, \xi_{d_u}) \tag{4.5}
\]

is called the ANOVA decomposition of $\theta(\xi)$, if

\[
\theta_0 = \int_{I_{d_u}} \theta(\xi) d\mu(\xi), \tag{4.6}
\]

and

\[
\int_I \theta_{j_1 \cdots j_k} d\mu(\xi_{j_k}) = 0 \quad \text{for} \quad 1 \leq k \leq s. \tag{4.7}
\]

We call $\theta_{j_1}(\xi_{j_1})$ the first-order term (or first-order component function), $\theta_{j_1,j_2}(\xi_{j_1}, \xi_{j_2})$ the second-order term (or second-order component function), etc.

The terms in the ANOVA decomposition are computed as follows:

\[
\theta_s = \int_{I_{d_u} - |S|} \theta(\xi) d\mu(\xi_{S^c}) - \sum_{T \subseteq S} \theta_T(\xi_T), \tag{4.8}
\]
where $S = \{j_1, j_2, \cdots, j_s\}$, $|S|$ is the number of elements in $S$, $T = \{i_1, i_2, \cdots, i_t\}$ is a subset of $S$, i.e. $\{i_1, i_2, \cdots, i_t\} \subset \{j_1, j_2, \cdots, j_s\}$ and $\theta_T = \theta_{i_1, i_2, \cdots, i_t}(\xi_{i_1}, \xi_{i_2}, \cdots, \xi_{i_t})$.

An important property of the ANOVA decomposition of $f$ is that the variance of $f$ is the sum of the variances of all the ANOVA terms except $\theta_0$:

$$
\sigma^2(\theta) = \sum_{s=1}^{d_u} \sum_{|S|=s} \sigma^2(\theta_S), \quad \sigma^2(\theta_S) = \int_{I^{d_u}} \theta_S^2 d\mu(\xi),
$$

or equivalently:

$$
\sigma^2(\theta) = \sum_{1 \leq j_1 \leq d_u} \sigma^2(\theta_{j_1}) + \sum_{1 \leq j_1 < j_2 \leq d_u} \sigma^2(\theta_{j_1, j_2}) + \cdots + \sigma^2(\theta_{1, 2, \cdots, d_u}).
$$

Computing the ANOVA decomposition, i.e. the constant term and high-order terms from Eqn. (4.6) and Eqn. (4.8) respectively, can be very expensive for high dimensional problems or complicated functions $\theta(\xi)$. One therefore uses the Dirac measure instead of the Lebesgue measure in integrations, i.e., $d\mu(\xi) = \delta(\xi - c)d\xi$, $c \in I^{d_u}$. The point “$c$” is called the “anchor point” and this method is called “anchored-ANOVA”. Then the (approximate) evaluation of the integral that appears in the first term of the right hand side of Eqn. (4.8) becomes much easier. For example, for the constant term and first-order term we have

$$
\theta_0 = \theta(c) \quad \text{(4.9)}
$$

$$
\theta_j(\xi_j) = \theta(c_1, \ldots, c_{j-1}, \xi_j, c_{j+1}, \ldots, c_{d_u}) - \theta_0, \quad j = 1, \ldots, d_u. \quad \text{(4.10)}
$$

Note also that Eqn. (4.8) implies that the $|S|$-order terms can be constructed recursively from all ANOVA terms whose orders are less than $|S|$.
For numerical purposes we approximate \( \theta(\xi) \) by all ANOVA terms whose degrees are less than or equal to \( \nu \):

\[
\theta(\xi) \approx \theta_0 + \sum_{j_1 \leq d_u} \theta_{j_1}(\xi_{j_1}) + \sum_{j_1 < j_2 \leq d_u} \theta_{j_1,j_2}(\xi_{j_1}, \xi_{j_2}) + \cdots + \sum_{j_1 < j_2 < \cdots < j_\nu \leq d_u} \theta_{j_1,j_2,\cdots,j_\nu}(\xi_{j_1}, \xi_{j_2}, \cdots, \xi_{j_\nu}).
\]  

(4.11)

Here \( d_u \) is called nominal dimension, and \( \nu \) is called the truncation or effective dimension. If \( \nu \) is low, then this type of approach, i.e. approximating the \( d_u \)-dimensional problem by a series of lower-dimensional problems, can greatly alleviate the computational burden. For example, let us consider the integration of the function \( \int \theta(\xi) d\xi \), e.g., \( \theta \) here can be the integrand in Eqn. (2.7), or the non-Galerkin integral to be seen in Eqn. (4.22). Since the integration is a linear operator, the integral can be approximated by the sum of integrals of ANOVA terms, i.e.

\[
\int_{I_{d_u}} \theta(\xi) d\xi \approx \int_{I_{d_u}} \theta_0 d\xi + \sum_{s=1}^{\nu} \sum_{j_1 < \cdots < j_s} \int_{I_{d_u}} \theta_{j_1,\cdots,j_s}(\xi_{j_1}, \cdots, \xi_{j_s}) d\xi.
\]  

(4.12)

Then, the \( d_u \)-dimensional integration problem becomes much lower dimensional (up to \( \nu \) assuming \( \nu \ll d_u \)) integration, where we can use collocation methods such as those involving Gaussian quadrature and weights. Consider the first-order term \( \theta_1(\xi_1) \) for instance. Let \( \mathbf{c}_{-1} = (c_2, c_3, \ldots, c_{d_u}) \) and \( (q^j_1, w^j)_{j=1}^{\mu} \) be the quadrature points and corresponding weights for integration along the first dimension, with \( \mu \) being the number of quadrature points. Then, the integration of \( \theta_1(\xi_1) \) can be approximated by

\[
\int_{I_{d_u}} \theta_1(\xi_1) d\xi_1 \approx \sum_{j=1}^{\mu} \theta_1(q^j_1) w^j = \sum_{j=1}^{\mu} (\theta(q^j_1, c_2, c_3, \ldots, c_{d_u}) - \theta_0) w^j.
\]  

(4.13)

See [34, 122] for more details. In [122], the authors applied the ANOVA method for a stochastic incompressible flow problem with a nominal dimension of parametric space up to 100 but with an effective dimension of 2 as an efficient dimension-reduction
technique. In §4.3 we demonstrate the use of ANOVA to approximately describe coupled neuronal networks that have multiple independent heterogeneous parameters.

4.3 Application to pre-Bötzinger complex neurons

Pre-Bötzinger complex model

We consider a network of model neurons previously studied as a model for rhythmic oscillations in the pre-Bötzinger complex [65, 91]:

\[
\begin{align*}
C \frac{dV_i}{dt} &= -g_{Na}m(V_i)h_i(V_i - V_{Na}) - g_l(V_i - V_l) + I^i_{syn} + I^i_{app}, \quad (4.14a) \\
\frac{dh_i}{dt} &= \frac{h_\infty(V_i) - h_i}{\tau(V_i)} \quad (4.14b)
\end{align*}
\]

for \( i = 1, ..., N \), where

\[
I^i_{syn} = g_{syn}(V_{syn} - V_i) \sum_{j=1}^{n_{nodes}} A_{i,j}s(V_j). \quad (4.15)
\]

Here \( V_i \) is the membrane potential of neuron \( i \), and \( h_i \) is a channel state variable for neuron \( i \) that governs the inactivation of persistent sodium. The first and second term of the right hand side in Eqn. (4.14a) is a persistent sodium current and passive leakage current, respectively and \( g_{Na}, V_{Na}, g_l, V_l \) are corresponding nominal parameters [13, 14]. Eqn. (4.14) was derived from the models in Butera et al. [13, 14] by blocking currents responsible for action potentials; Rubin [90] considered a similar model with \( n_{nodes} = 2 \), and Dunmyre and Rubin [31] considered synchronization in the case \( n_{nodes} = 3 \). The
model equations use four specifically-parameterized sigmoid functions,

\begin{align}
  s(V) &= \frac{1}{1 + \exp(-(V + 40)/5)}, \quad (4.16) \\
  \tau(V) &= \frac{1}{\epsilon \cosh((V + 44)/12)}, \quad (4.17) \\
  h_\infty(V) &= \frac{1}{1 + \exp((V + 44)/6)}, \quad (4.18) \\
  m(V) &= \frac{1}{1 + \exp(-(V + 37)/6)}. \quad (4.19)
\end{align}

The functions \(\tau(V), h_\infty(V)\) and \(m(V)\) are a standard part of the Hodgkin-Huxley formalism [43], and synaptic communication is assumed to act instantaneously through the function \(s(V)\). The neurons are coupled through a synaptic current \(I_{syn}^i\) for \(g_{syn} \neq 0\) where \(A_{ij}\) is a symmetric adjacency matrix, i.e. \(A_{ij} = 1\) if neuron \(i\) and \(j\) are connected, and \(A_{ij} = 0\) otherwise. A previous study considered only all-to-all coupled networks [65], but we will consider a more structured network below. We denote the degree of \(i\)-th neuron (its number of neighbors) by \(\kappa_i\), i.e. \(\kappa_i = \sum_{j \neq i}^{n_{\text{nodes}}} A_{ij}\).

It was shown in [65, 91] that if the values of the applied currents \(I_{app}^i\) are uniformly distributed in a certain interval, synchronous behavior is observed after a transient, i.e. all neurons oscillate periodically with the same period, although the heterogeneity in the \(I_{app}^i\) means that each neuron follows a slightly different periodic orbit in its own \((V, h)\) phase space. It appears that (asymptotically in time) the values of the \(V_i\) and \(h_i\) vary smoothly as a function of the heterogeneous parameter \(I_{app}^i\). This observation leads to the continuum limit of Eqn. (4.14):

\begin{align}
  C \frac{\partial V(\mu, t)}{\partial t} &= -g_{Na}m(V(\mu, t))h(\mu, t)(V(\mu, t) - V_{Na}) - g_l(V(\mu, t) - V_l) + I_{syn} + I_{app} \quad (4.21a) \\
  \frac{\partial h(\mu, t)}{\partial t} &= \frac{h_\infty(V(\mu, t)) - h(\mu, t)}{\tau(V(\mu, t))} \quad (4.21b)
\end{align}

55
where $I_{app}$ is parameterized as $I_{app} = I_m + I_s \mu$ with $\mu$ being a uniform distribution on $[-1, 1]$, i.e. $I_{app}$ follows a uniform distribution on $[I_m - I_s, I_m + I_s]$ and

$$I_{syn}(\mu,t) = g_{syn}(V_{syn} - V(\mu, t)) \int_{-1}^{1} s(V(\mu, t)) p(\mu) d\mu.$$ (4.22)

Note that $p(\mu)$ is a probability density function for $\mu$, i.e. $p(\mu) = 1/2$ for $-1 \leq \mu \leq 1$. In this limit $V_i(t)$ and $h_i(t)$ become the functions $V(\mu, t)$ and $h(\mu, t)$, respectively. The results for $n_{nodes} \to \infty$ should provide a good approximation to the behavior seen when $n_{nodes}$ is large but finite, as we expect it to be. Rubin and Terman [91] first introduced the continuum limit, their contribution being largely analytical. Laing et al. [65] presented a computationally efficient way to describe the heterogeneous network by applying techniques widely used in the uncertainty quantification (UQ) community known as generalized Polynomial Chaos and the associated stochastic collocation method [120, 121, 41], as discussed in §2.2. These methods are high-order accurate, in fact exponentially accurate, but suffer when the dimensionality of the parametric space increases; this is known as the so-called curse of dimensionality. Sparse grids techniques have greatly alleviated this problem by utilizing the smoothness of the function in low to moderate “heterogeneity dimensions” [33, 40]. However, the complexity estimate of sparse grids still depends heavily on the dimension and on the regularity of the functions being integrated. To push the dimensionality barrier higher, several methods have been introduced in the UQ community; one of them is the ANOVA method, which will be described below for a case in which there are multiple heterogeneous physiological (intrinsic to each neuron) parameters. In the sections 4.1 and 4.2 we briefly review the PCE and ANOVA methods; see [33, 121, 122, 41] for more details.
Case 1: Multiple heterogeneous parameters

We consider the case where there exist four heterogeneous parameters: $I_{app}, g_{Na}, V_{syn},$ and $V_{Na}$ are all independently and uniformly distributed. Each of these four parameters can be parameterized by their mean and half-width, together with the standard uniform distribution $\xi_i, i = 1, 2, 3, 4,$ which we denote by $\xi_i \sim U[-1, 1],$ and whose probability distribution function is $p(\xi_i) = \frac{1}{2}$ for $-1 \leq \xi_i \leq 1.$ For example, if $I_{app} \sim U[17.5, 32.5],$ then it is parameterized as $I_{app} = \mathbb{E}[I_{app}] + h(I_{app})\xi$ where $\mathbb{E}[I_{app}] = 25$ and $h(I_{app}) = 7.5$ are the mean and half-width of $I_{app},$ respectively, and $\xi$ has the standard uniform distribution. Then, as mentioned in the above section, the continuous variables $V$ and $h$ become a function of these $\xi_i$'s as well as time $t$ as $V(t; \xi_1, \xi_2, \xi_3, \xi_4)$ and $h(t; \xi_1, \xi_2, \xi_3, \xi_4),$ respectively and the sum in Equation (4.15) is represented by the integral

$$\int_{\Omega^4} s(V(t; \xi))p(\xi)d\xi_i \quad (4.23)$$

where $\xi = (\xi_1, \xi_2, \xi_3, \xi_4),$ $p(\xi) = \prod_{i=1}^4 p_i(\xi_i),$ and $\Omega = [-1, 1].$ In stochastic collocation or sparse grid methods this integral is approximated as the sum of the function evaluated at the collocation points multiplied by their corresponding weights; see [65, 120] for more detail. In ANOVA methods, we first approximate the function $s(V(t; \xi))$ by its ANOVA terms whose orders are less than $\nu$ as in Eqn. (4.11). Then the integral of a high-dimensional function is represented by the integral of a series of low-order functions, which can be easily computed by standard numerical integration techniques. For example, assume that $\nu = 1.$ The ANOVA approximation of $s$ denoted by $s_A$ is then as follows:

$$s(V(t; \xi)) \approx s_A(V(t; \xi)) = s_0 + \sum_{j=1}^4 s_j(\xi_j) \quad (4.24)$$

where $s_j(\xi_j)$ is given in Eqn. (4.10). For example, for $j = 2,$ $s_2(\xi_2) = s(V(t; c_1, \xi_2, c_3, c_4))$ for an anchor point $c = (c_1, c_2, c_3, c_4).$ Then the integral in Equation (4.23) is com-
puted as the sum of the integral of the constant term and the first-order ANOVA terms, which are readily computable:

$$
\int_{\Omega^4} s(V(t; \xi))p(\xi)d\xi_i \approx \int_{\Omega^4} s_A(V(t; \xi))p(\xi)d\xi_i = \mathbb{E}[s_0] + \sum_{j=1}^{4} \mathbb{E}[s_j(\xi_j)] \quad (4.25)
$$

where $\mathbb{E}[f]$ is the expectation operator of $f$ with respect to the probability measure $p(\xi)$.

All four heterogeneous parameters here are drawn from a uniform distribution: $I_{app}$ on $[17.5, 32.5]$, $V_{syn}$ on $[-1,1]$, $V_{Na}$ on $[49,51]$, and $g_{Na}$ on $[2.55, 3.05]$. The other parameters are given as follows:

$$
V_{Na} = 50, \quad V_{syn} = 0, \quad g_{syn} = 0.3, \quad g_l = 2.4, \quad V_l = -65, \quad \varepsilon = 0.1, \quad C = 0.21.
$$

The parameters for sparse grids and ANOVA are shown in Table 4.1. We also consider a direct Monte Carlo (MC) method with 10,000 points (i.e. 10,000 all to all coupled neurons) as a reference solution. For each of sparse grids, ANOVA, and MC, given the sampling (or collocation) points, we solve deterministic problems. In this sense, these methods might be said to be nonintrusive. However, as the manner of computing the coupling integral of Eqn. (4.22) depends on the scheme chosen, we label this section as “intrusive” (or perhaps semi-intrusive) in contrast with the method of the next section. Fig. 4.1 shows the behavior of the $V_i$ and $h_i$ corresponding to 10 samples from the sparse grids in Table 4.1.

First we solve Eqn. (4.21a) and Eqn. (4.21b) for $V$ and $h$ using sparse grids, ANOVA and MC methods and compare the mean and variance of $V$ and $h$ derived from the three methods. For example, given sparse grid points and corresponding weights
Figure 4.1: Solutions of Eqn. (4.14a) and Eqn. (4.14b) when there are four heterogeneous parameters and samples come from sparse grids (top) and ANOVA (bottom) with parameters given in Table 4.1. Left: $V_i$ as functions of time. Right: $h_i$ as functions of time. Different line styles correspond to different neurons (only ten neurons are shown) and they show that neurons with different parameters behave differently.
Table 4.1: In sparse grids, the number of collocation points is determined by the level, i.e. the higher the level the more points. In the ANOVA method, $\mu$ is the number of collocation points in one direction and $\nu$ is the truncation dimension of the ANOVA decomposition, i.e. $\nu = 2$ means that we consider only the first and second-order interaction terms. For these parameters 411 points are needed for the sparse grid method and 171 points for the ANOVA method.

<table>
<thead>
<tr>
<th>configuration</th>
<th>Sparse Grid</th>
<th>ANOVA</th>
</tr>
</thead>
<tbody>
<tr>
<td>level=3</td>
<td>411</td>
<td>$\mu = 5, \nu = 2$</td>
</tr>
</tbody>
</table>

For each of the sets $\{\xi^{(j)}, w^{(j)}\}_{j=1}^{n_{\text{nodes}}}$, the mean and variance for $V$ can be computed as

$$
\mathbb{E}[V](t) = \sum_{j=1}^{n_{\text{nodes}}} V(t; \xi^{(j)})w^{(j)}
$$

$$
\mathbb{V}[V](t) = \sum_{j=1}^{n_{\text{nodes}}} V^2(t; \xi^{(j)})w^{(j)} - \mathbb{E}[V]^2(t)
$$

where $V(t; \xi^{(j)})$ is the solution to Equation (4.21a) with $\xi = \xi^{(j)}$. Fig. 4.2 and Fig. 4.3 show the mean and variance for $V$ and $h$, respectively, calculated using the three methods, and the results agree well with one other. Note that they are visually indistinguishable but when zoomed in (inset figure), a slight difference can be perceived between MC and the other two methods. This strongly suggests that the ANOVA method can help model high-dimensional heterogeneous parametric problems, in addition to its extensive use in high-dimensional uncertain parametric problems.

Based on this observation, we describe a low-dimensional system using only the ANOVA method from now on in this subsection.

**Coarse Dynamics and Stability.** We will now consider the PCE coefficients $\alpha_i$ and $\beta_i$ (with $i = 1, 2, \ldots, n_{\text{coeffs}}$) for $V$ and $h$, respectively as our reduced, coarse-grained variables, i.e. we approximately represent $V$ and $h$ as

$$
V(t; \xi) = \sum_{i=1}^{n_{\text{coeffs}}} \alpha_i(t)\psi_i(\xi)
$$

$$
h(t; \xi) = \sum_{i=1}^{n_{\text{coeffs}}} \beta_i(t)\psi_i(\xi)
$$

(4.26)
where each $\psi_i(\xi), i = 1, ..., n_{\text{coeffs}}$ is a product of Legendre polynomials of the variables in $\xi = \{\xi_1, \xi_2, \xi_3, \xi_4\}$. We explore the long-term dynamics of (4.14) using these coarse-grained variables and compute PCE coefficients using ANOVA methods, as there are four heterogeneous parameters.

Figure 4.2: The mean (4.2(a)) and variance (4.2(b)) for $V$. MC simulations with 10,000 points are considered as the reference solution. Note that results from all methods are visually indistinguishable.

Figure 4.3: The mean (4.3(a)) and variance (4.3(b)) for $h$. MC simulations with 10,000 points are considered as the reference solution. Note that all methods are visually indistinguishable.
Equation-Free Computations. Availability of the governing equations for the variables of interest is a prerequisite to modeling and computation. However, if the underlying differential equations are nonlinear or nontrivial and $\xi$ is high-dimensional, then the right hand side in Eqn. (4.3c) is often coupled and very complicated making it almost impossible to obtain it in explicit, closed form. We circumvent this step using the equation-free (EF) framework for complex, multiscale systems modeling [56, 109]. In this framework we can perform system-level computational tasks without explicit knowledge of the coarse-grained equations. This is accomplished through the operators that transform between coarse and fine variables. The mapping from coarse to fine variables is called the lifting operator ($L$) while the mapping from fine to coarse variables is called the restriction operator ($R$).

We denote the detailed (fine), microscopic time-evolution operator for the ODEs of Eqn. (4.14), as defined in Eqn. (2.1), by $\Phi_{\tau,F}$ (where $\tau$ represents the number of time steps or iterations). The macroscopic evolution operator $\Phi_{\tau,C}$ can then be defined as in Eqn. (2.1) as:

$$\Phi_{\tau,C}(\alpha(t)) = R \circ \Phi_{\tau,F} \circ L(\alpha(t))$$

(4.27)

where $\alpha(t) \in \mathbb{R}^{n_{\text{coarse}}=2n_{\text{coeffs}}}$ is the vector of PCE coefficients ($\alpha_1, ..., \alpha_{n_{\text{coeffs}}}, \beta_1, ..., \beta_{n_{\text{coeffs}}}$) in Eqn. (4.26); i.e. the coarse-grained variables. The general procedure consists of five steps; (i) identifying good coarse-grained variables $\alpha$, (ii) constructing a lifting operator that maps the coarse variables to one or more consistent fine scale realizations, (iii) evolving the fine scale equations for certain amount of time, (iv) restricting the resulting fine variables to the coarse variables in order to estimate their time derivatives, and (v) repeating the procedure to perform specific computational tasks.

We first demonstrate coarse projective integration (CPI) [38]. The PCE coefficients $\alpha_i$ and $\beta_i, i = 1, 2, \ldots, n_{\text{coeffs}}$ for $V$ and $h$ in Equation Eqn. (4.26) are considered as the coarse-grained variables and obtained via Eqn. (2.9). For comparison, we also
evolve the detailed (fine) coupled equation (Eqn. (4.21)) from which we record the coefficients (coarse-grained variables) at every time step. The forward Euler method with a fixed step size of 0.001 is used as a time integrator. For CPI, the detailed (fine) coupled system Eqn. (4.21) is integrated forward in time using short bursts of fine-scale simulations consisting of 7 steps. Then, the coarse variables \( \alpha \) are evaluated according to Eqn. (2.7), where the integral is computed by the ANOVA method given in Eqn. (4.12). The last few observations of the coarse variables \( \alpha \) are used to estimate their (coarse) time-derivative. Finally we integrate the coarse variables with a forward Euler jump of 7 steps, thus save 7 inner integration steps at every 7 steps. Fig. 4.4 shows the second and third PCE coefficients from coarse projective integration and from full detailed simulation, and shows that they agree well with each other. For the given parameters, in particular with \( \mathbb{E}[I_{app}] = 25 \), the network exhibits stable, synchronized periodic behavior as shown in Fig. 4.5.

Figure 4.4: Coarse projective integration (dashed lines) and detailed (fine) coupled dynamics (solid lines) for \( V \) (left) and \( h \) (right). Two PC coefficients \((\alpha_1, \alpha_2)\) and \((\beta_1, \beta_2)\) are shown for \( V \) and \( h \), respectively. Forward Euler with a fixed step size of 0.001 is used as a time integrator. For coarse projective integration, it jumps with a forward Euler of 7 step after estimating time derivatives.

The equation-free approach is also useful for computing long-time (stationary) states and their stability and dependence on parameters \[56, 71\]. The coarse time-
Figure 4.5: Left: Periodic orbit of the mean of $V$ and the mean of $h$ when $E[I_{app}] = 25$. Coarse projective integration (dashed) and detailed (solid) simulation. Temporal profile of $E[V]$ (top right) and $E[h]$ (bottom right) corresponding to one period of limit cycle.

stepper $\Phi_{\tau,C}(\alpha(t))$ is defined as mapping from $\alpha(t)$ to $\alpha(t + \tau)$ via one iteration of the equation-free method as mentioned in the above: lifting a coarse-grained initial condition $\alpha(t)$ to one or more consistent fine initial conditions, integrating the full (fine) model for a (short) time $\tau$, and then restricting to the coarse observable of the final fine state $\Phi_{\tau,C}$. In order to compute the stationary states we solve for the fixed point $\alpha^*$ satisfying

$$F_\tau(\alpha) \equiv \Phi_{\tau,C}(\alpha) - \alpha = 0,$$

which is referred to as the coarse flow map (as previously defined in Eqn. (2.3)). Iterative matrix-free linear algebra algorithms such as Newton-GMRES can be used to find zeros of such a function in the absence of explicit equations for the dynamics of the coarse variables $\alpha$. Eigenvalues of the Jacobian of the coarse flow map $F_\tau$
evaluated at a fixed point reveal the (coarse grained) stability of that fixed point and help determine the nature of its potential bifurcations. Fig. 4.6 shows the first 10 eigenvalues of the Jacobians of both the fine and coarse flow maps at equivalent fixed points. As the polynomial degree in the coarse flow map (the number of coarse variables) increases, these coarse eigenvalue estimates are expected to approach the leading eigenvalues of the Jacobian of the fine flow map, and this is clearly seen in Fig. 4.6.

Figure 4.6: Eigenvalues of the Jacobian of the fine flow map and the coarse flow map at corresponding fixed points, obtained with three different PCE orders: $P = 1$ (leading to 10 coefficients), $P = 2$ (30 coefficients), and $P = 3$ (70 coefficients). As the polynomial degree -and thus the number of coarse variables- increases, the eigenvalues from the coarse flow simulation show increasingly better agreement with those from the fine simulation.
Case 2: Intrinsic and structural heterogeneities

We consider Eqn. (4.14) with the following physiological parameter values [65]

\[ V_{Na} = 50, \quad V_{syn} = 0, \quad g_{syn} = 0.3, \quad g_l = 2.4, \quad V_l = -65, \quad \varepsilon = 0.1, \quad C = 0.21. \]

For a heterogeneous network, \( I_{app} \) is chosen to follow a uniform distribution on [17.5, 32.5], parameterized by \( I_{app} = 25 + 7.5\omega \) where \( \omega \) is uniformly distributed on \([-1, 1]\). Neurons are connected in a Chung-Lu type network [64] defined as in §3.2 (see Eqn. (3.6)). For this section, we choose \( n_{nodes} = 512, p = 0.5, \) and \( r = 0.1 \). For Eqn. (3.5), we use \( q = (n - i)/(i - 1) \).

Clearly, we can consider the way the neurons are connected in the network a a structural heterogeneity (as in chapter 3), as opposed to the intrinsic heterogeneity used earlier in this chapter. In our case, we assume that this structural heterogeneity is well described by the degree distribution: the degree of each neuron denoted by \( \kappa \) is the important structural heterogeneous parameter, and its probability distribution is the degree distribution of our Chung-Lu network.

For these parameter values, and a particular realization of a Chung-Lu network with 512 neurons, we observe that the network eventually synchronizes, and all neurons evolve along a periodic trajectory (each in a slightly different periodic path, since the neurons differ both intrinsically and in their connectivities). At any point in time, the state at each neuron, \( (V_i, h_i)(t) \) can be approximated by a smooth surface in two heterogeneous parameters \( \xi = (\kappa_i, I_{app,i})_{i=1}^{n_{nodes}} \) according to Eqn. (4.26). Though these are parameters in the sense that they do not change in time, they are still unique for each neuron. If indeed the behavior can be expressed as a function of our two heterogeneous parameters and time, this suggests that at every moment in time the values of the dynamic variables of each neuron would lie on a smooth surface, here a two dimensional one, parametrized by the two measures of heterogeneity. At every
point in time the 512 individual variable values, one for every neuron, would lie on, or very close to, this surface.

Fig. 4.7 shows the potential $V_i$ of all the neurons for $0 \leq t \leq 4$ and the evolving “heterogeneity surface” of the potential $V_i$ at two instances in time $t = 1.27$ and $3.55$ (marked on the figure) as a function of the two heterogeneous parameters, which are randomly picked at on a limit cycle. The fact that, for all practical purposes, the values of the variables for each neuron lie on, or close to such a smooth surface, implies that a PCE representation performs well as a coarse-grained descriptor of the heterogeneous neuronal population.
$V$ of each neuron (represented by colored filled circles), clearly lying on, or very close to, the smooth two-dimensional surface of the coarse-grained description.

Figure 4.8: Limit cycles of the $(V_i, h_i)$ for $n_{\text{nodes}} = 512$ neurons. Each filled circle represents the potential of one neuron, with different colors denoting different time snapshots along the synchronized oscillation. The oscillations proceed in the clockwise direction, and the surfaces in the insets show the $V_i$ as functions of the two heterogeneous parameters at nine different times cut.

4.4 Conclusion

We have proposed and demonstrated the use of several distinct forms of dimension reduction for the computationally efficient study of heterogeneous networks of coupled neurons. In Case 1 we considered an all-to-all coupled network with four independent heterogeneous parameters. To efficiently simulate such a network we need to approximate a four-dimensional integral, which we accomplished using ANOVA methods. A reduced model of this type of network can also be formulated using coefficients in a polynomial chaos expansion in the heterogeneous parameters as the "coarse" variables. Having such a reduced model leads to an improvement in the speed for a variety of
computations of interest (direct simulation, coarse limit cycle computation, coarse stability analysis) which we demonstrated using the equation-free framework.

In Case 2 we considered a network with both intrinsic and structural heterogeneity, and showed that we could expand the state variables in polynomials of both the intrinsically varying parameter and a feature of the network connectivity—in this case, the degree of each neuron. To do this, we need to construct orthogonal polynomials with respect to the network degree distribution. If this (integer) distribution is known 

\textit{a priori}, then the polynomials can be found in the literature [119], or easily constructed using the recurrence relation [116]. If the distribution is unknown, and we only have samples of it available, then the convergence of the “empirical” polynomials based on the sampled distributions, to the “true” distributions at the limit of infinite neurons becomes an interesting research problem that we are currently investigating. We believe that all these approaches can play an important practical role in accelerating the computational study (and, in general, the modeling) of complex heterogeneous networks, and we are exploring the practical limits of (a) how many independently distributed heterogeneous parameters one can usefully approximate and (b) the modeling of heterogeneities that are not independently distributed, but rather exhibit correlations.
Chapter 5

Data-driven representations: an “equal” embedding space

In this chapter, we turn our focus to discovering a space akin to the physical space used partial differential equations (PDEs) given only discretely-sampled trajectories. Such trajectories could be recordings at fixed locations in space, sampling a truly continuous medium, or they could be the output of a spatially discretized simulation of a PDE, or, of most interest here, they could be similar data for a system with no obvious spatial extent. In each case, we apply diffusion maps to uncover the space in question as a dimension reduction of the population of individual time series.

This approach can be compared to reassembling shredded document—if sufficient similarity is apparent between adjoining strips, it is possible to reassemble the original order of the strips. Such an interpretation is suggested by Fig. 5.1(a). Here, the page to be shredded is the printout of a simulation of a nonlinear partial differential equation (a one-dimensional Complex Ginzburg-Landau [35]). Spatiotemporally intermittent patterns of the complex amplitude are plotted as a function of time and space (1000

This work has been submitted for publication. [52]
snapshots in the vertical and 512 space “pixels” in the horizontal direction). Shredding the page and randomly shuffling the resulting columns (time series of the amplitude for each of the 512 spatial locations) leads to something visibly uninformative (Fig. 5.1(c)). Yet, as we will see below, one can use nonlinear data mining techniques to put the page back together again. Indeed, quantitative pairwise comparisons of the temporal patterns on each strip can recover each strip’s nearest neighbors, ultimately putting the page back together again (though, since the method relies on diffusion maps, which operates through an eigendecomposition, results are only unique up to multiplication by a possibly negative constant, resulting in our reconstruction being the mirror image of the original data). The implication is that even when we do not know where our sensors (the strips) are located in space, we can put them in the right order solely through what they recorded - spatial information was, in a sense, implicit in the temporal measurements.

The extraction of such useful information implicit in the apparently disorganized data is the subject of this chapter. Using the intrinsic variability in the recorded data, we will show how to construct suitable coordinate sets (suitable embedding spaces) in which to plot, visualize, observe, hopefully understand and even model the data.

When constructing this space, we do not alter the data nor the information we can extract from the data, making it a space equal to the space in which the observer originally record the measurements, suggesting the name ”Equal Space“.

In recent years, many nonlinear manifold learning techniques have been proposed to embed nonlinear phenomena in a low dimensional way. Examples include isomap [108], locally linear embedding [89] and Laplacian eigenmaps [8]. The method of our choice is diffusion maps [21, 101, 25, 18]. Although we restrict our analysis using a simple Euclidean distance kernel, many extensions exist, such as vector diffusion maps [98] or diffusion maps using more refined distance metrics, e.g. the Mahalanobis distance [100].
In this chapter, we demonstrate that diffusion maps are able to extract the variabilities contained in the data from a variety of qualitatively different situations.

First, by considering time series at different points in space, we find that, on the finest scale, the spatial coordinate is the dominant variability in the data, and thus the dominant equal-space coordinate.

The problem involving physical space is then expanded to more general situations: in a system that does not possess physical spatial extent, i.e. a network, we demonstrate that the same nonlinear data mining tools are capable of discovering informative coordinates which span the equal space. This exemplifies that this space is, in general not spanned by the physical spatial axes, but rather by other intrinsic variabilities,
such as variations in the behavior of the individual entities, which are implicitly contained in the data and which we extract with our algorithm. These variabilities depend on the coarsening scale we introduce in our algorithm.

We conclude with a discussion of what we see as potential advantages (and some limitations) of the approach, both for systems with physical and without physical spatial extent. Details to the individual models and to the simulation methods are given in the Supplementary Information [52].

5.1 Recovering space from spatiotemporal data

![Figure 5.2](image)

Figure 5.2: (a) Spatiotemporal data obtained from numerical simulations of the complex Ginzburg-Landau equation with two spatial dimensions. The data is colored with the real part of the complex amplitude $W$. (b) The first two independent diffusion modes $\phi^{(1)}$ and $\phi^{(2)}$, colored as in (a).

We start by investigating a spatiotemporal chaotic phenomenon in a system with one spatial dimension. In particular, we consider spatiotemporal intermittency in the complex Ginzburg-Landau equation, a general reaction-diffusion equation for oscillatory systems with a spatial extension [97, 15, 35]. Characteristic for these chaotic dynamics is the repeated appearance of synchronous patches at seemingly random places in space and time. Following their emergence, these patches shrink with a constant rate due to diffusion, giving rise to triangular patterns, see Fig. 5.1(a).
Note that due to the incoherent nature of the dynamics, the individual time series of a discretized version of this dynamical system are mutually different. Nevertheless, due to the diffusion in the system, trajectories located close to each other in space tend to be similar. This fact can be exploited in the diffusion maps approach, where the Euclidean distance between neighboring time series is expected to be small compared to the distances between time series separated by a larger distance in physical space.

By calculating the diffusion matrix, it can be shown that the first diffusion mode $\phi^{(1)}$ correlates with the spatial dimension $x$: each entry of $\phi^{(1)}$ corresponds to a particular time series, and if any one entry is large, the corresponding time series is located at large $x$, while for small entries, the trajectories are located towards the left end of the spatial axis, see Fig. 5.1(b).

Note that here, the kernel scale $\epsilon \approx 0.024D_{\text{max}}$ is chosen very small, such that only a few nearest neighbors are considered in the diffusion maps approach. Moreover, by investigating not only the first but also the following diffusion modes, one finds that $\phi^{(1)}$ is the only independent direction, indicating that there exists only one spatial dimension in the system. This enables us to actually find the spatial order of the data set, as can be shown by shuffling the individual time series, c.f. Fig. 5.1(c), before calculating the diffusion modes of the system. The first and only independent direction is then found to be nonmonotonic, as shown in Fig. 5.1(d). However, by sorting the entries of the diffusion mode in increasing order and then applying this arrangement to the corresponding time series, the spatial arrangement is recovered, see Fig. 5.1(e-f). Thus, we argue that the physical space is an intrinsic heterogeneity in this system. Nevertheless, the diffusion maps outcome is not unique. Due to the symmetry of the approach, the diffusion mode can either correspond to $x$ or to $-x$. That is, it either parametrizes the left-to-right direction or vice versa. This can be observed in Figs. 5.1(e) and (f), where the sorted eigenvector $\phi^{(1)}$ correlates with $-x$. 

74
and thus sorts the data in a mirrored way. Comparing the pictures Fig. 5.1(a) and (e), we observe that they correspond except of a left-right flip along the $x$-axis.

Note that the dynamics are not only incoherent in space, but also in time. Hence, the individual snapshots are, as the time series, mutually different. By comparing the snapshots, as indicated by the horizontal lines in Fig. 5.1(a) instead of the time series, one can show that the time is also a heterogeneity in the system[52]. Furthermore, it is not surprising that by taking spatiotemporal patches, one can arrange them on a two dimensional plane, spanned by the variabilities space and time.

For a system with two spatial axes $x$ and $y$, the data of spatiotemporal intermittency obtained with zero-flux boundaries is depicted in Fig. 5.2(a). By calculating the distances between the time trajectories and using the diffusion maps approach, one finds two independent modes $\phi^{(1)}$ and $\phi^{(2)}$. When coloring these modes with $W$, one finds that the two modes indeed span a space that is one-to-one with the original physical space, as visually verified in Fig. 5.2(b). Note that the $\phi^{(i)}$s contain $N_x \cdot N_y$ entries, that is, time series, with $N_x$ and $N_y$ being the respective number of grid points in the $x$- and $y$-direction, respectively. Therefore, when calculating the distances between the time series, we lose the spatial information inherent in the system. Nevertheless, due to the similarity of nearby time series, diffusion maps finds that these are parametrized by two dimensions, and yields their actual two-dimensional order.

### 5.2 Recovering parameter space

Unlike the complex Ginzburg-Landau equation and other reaction-diffusion systems, there are systems of oscillators that do not posses a spatial arrangement. Examples are globally coupled systems or networks with random connections. In this section we investigate a neural network, where each node of the network is modeled as a neuron
from the pre-Bötzinger complex [91, 62, 16]. These Hodgkin-Huxley-type neurons oscillate periodically, whereby the state of each neuron is specified via its membrane potential $V$ and a channel state variable $h$. As in previous studies [16], we take a current variable of each neuron, $I_{app}^i$, as a heterogeneous parameter intrinsic to each neuron. The neurons are connected in a Chung-Lu type network [64, 52]. The number of connections to a neuron $i$ is the degree $\kappa_i$, which varies strongly between the neurons. That is, $\kappa_i$ is another, structural heterogeneous parameter. The temporal evolution of a network of 1024 neurons is depicted in Fig. 5.3(a). There, one observes that the neurons oscillate clustered in the $(V, h)$ plane, but with somewhat different phases and amplitudes. Due to the variabilities, their instantaneous values differ, c.f. Fig. 5.3(b). Nevertheless, it has been shown that the dynamics of all neurons can be described as a smooth function of two the heterogeneous parameters $I_{app}^i$ and $\kappa_i$ [16]. This is indicated by the color-code in Figs. 5.3(b-c). Note that there is a strong dependence of the dynamics on the intrinsic parameter $I_{app}^i$, whereas the dependence on the degree $\kappa_i$ is rather weak.

The parameter dependence in the network can also be recovered using diffusion maps. Calculating the pairwise distances between the time series and extracting the diffusion modes, yields that the first two diffusion modes parametrize a two-dimensional manifold. Hereby, $\epsilon = e^{10} \approx 93D_{\text{max}}$ was chosen large compared to the maximal distance in the data. By coloring the two diffusion modes with the heterogeneous parameter $I_{app}^i$, one observes a one-to-one correspondence between this parameter and the first mode, as depicted in Fig. 5.3(d). The second direction, perpendicular to the first mode, correlates with the degree $\kappa_i$, see Fig. 5.3(e). Note that the first few modes following $\phi^{(2)}$ are harmonics, indicating that there are only two major directions that parametrize the dynamics. Comparing these results with the outcome from §5.1, an analogy between the two parameters in the heterogeneous neural network and the spatial axes in the reaction-diffusion system becomes apparent.
Figure 5.3: (a) Temporal evolution of the two variables $h$ and $V$ of the individual oscillators in the Chung-Lu type network of pre-Bötzinger neurons. $V$ of the snapshots, indicated by the dots, are plotted as a function of the two first diffusion modes $\phi^{(1)}$ and $\phi^{(2)}$ at different time steps, as indicated in the figure. The smooth dependence of $V$ on the two diffusion modes is described with a least-squares fit of a sum of orthogonal polynomials. The 2D polynomials are the product of 1D polynomials obtained obtained by a method equivalent to Gram-Schmidt [79] using inner products weighted by the marginal distributions of the $\phi^{(1)} \times \phi^{(2)}$ data. (b) Each oscillator as a function of the two heterogeneous parameters $\kappa_i$ and $I_{i_{\text{app}}}$, colored with the voltages $V$ at $t = 0$. (c) Each oscillator as in (b), but colored with the variable $h$ of the snapshot depicted in (b). (d) The two independent diffusion modes $\phi^{(1)}$ and $\phi^{(2)}$, colored by applied heterogeneous current $I_{i_{\text{app}}}$. (e) The two diffusion modes $\phi^{(1)}$ and $\phi^{(2)}$ as in (d), colored by degree $\kappa_i$. Hereby, $\epsilon$ is taken as $\epsilon = e^{10} \approx 93D_{\text{max}}$. 

77
It is worth mentioning that the diffusion maps approach enables us, even without knowing the spatial dimensions or the heterogeneous parameters in our system, to extract the dominant variabilities on which the dynamics depend.

5.3 Tuning the scale parameter $\epsilon$

Figure 5.4: (a) Temporal evolution of the type II chimera state in a globally coupled version of the complex Ginzburg-Landau equation with one spatial dimension $x$ and periodic boundary conditions. The pseudo-color corresponds to the modulus of the complex amplitude $W$. (b) The first two independent diffusion modes $\phi^{(1)}$ and $\phi^{(2)}$ for $\epsilon = e^{-2.5} \approx 1.7 \cdot 10^{-3} D_{\text{max}}$, colored with the position $i$ along the spatial coordinate $x$. (c) The first two independent diffusion modes $\phi^{(1)}$ and $\phi^{(2)}$ for $\epsilon = e^7 \approx 22.5 D_{\text{max}}$, colored with the position $i$ along the spatial coordinate $x$. (d) First independent diffusion mode $\phi^{(1)}$ for $\epsilon = e^{-2.5} \approx 1.7 \cdot 10^{-3} D_{\text{max}}$ and $\epsilon = e^7 \approx 22.5 D_{\text{max}}$.

In the example of the spatiotemporal chaos and the heterogeneous neural network, we used a specific choice of the kernel scale $\epsilon$. In order to recover the spatial arrangement, we had to choose it very small, thereby considering only the nearest neighbors, that is, only very similar time series. In the case of the network, however, a larger $\epsilon$ revealed the dominant variabilities. In this section, we vary $\epsilon$ in order to extract different features of our data. As a model example, we investigate a chimera state, that is, a dynamical hybrid state of coexisting coherence and incoherence [61, 5, 83, 53]. An example of such a state can be observed in a globally coupled version of the complex Ginzburg-Landau equation. A simulation with one spatial dimension is depicted in Fig. 5.4(a). Note that here we take periodic boundaries, so that the
spatial axis is in fact a ring. This chimera state, also called type II chimera, has an underlying two-cluster state, in which one of the two clusters develops incoherent dynamics while the other cluster remains synchronized [96]. By choosing the kernel scale $\epsilon = e^{-2.5} \approx 1.7 \times 10^{-3} D_{\text{max}}$ very small compared to the maximal distance contained in the data, that is, again considering only very similar time series, we are able to reconstruct the spatial arrangement, as depicted in Fig. 5.4(b). Note that due to the periodicity of the spatial axis $x$, two diffusion modes are needed to embed the data. Furthermore, it is worth mentioning that the approach maps the coherent oscillations, which are, due to their synchrony, very similar, onto a very concentrated cluster in diffusion space. Nevertheless, by zooming in on this cluster, we find that the two diffusion modes are still able to differentiate between the synchronous series, see inset of Fig. 5.4(b). This is possible since the diffusion in the system still preserves a slight drift in the coherent cluster and therefore allows for a discrimination of the coherent time series.

Tuning the scale parameter in the diffusion kernel alters the decay rate of the Gaussian function, practically allowing for the specification of how many ‘similar’ time series shall be considered by the diffusion maps algorithm. When using a very large $\epsilon$, one therefore obtains a rather coarse description of the data.

For the network of pre-Bötzinger neurons, a rather large $\epsilon$ revealed the heterogeneous parameters $I_{\text{app}}^i$ and $\kappa_i$ in the system. This can be explained by the fact that the heterogeneous parameters influence the overall coarse dynamics, but, in contrast to the spatiotemporal intermittency, points with similar variabilities are not necessarily similar in the sense that their mutual Euclidean distances are small, but rather their distances to all other points are similar. They have, so to say, similar coarse behavior. Thus, we in general expect to obtain the coarse variabilities in our data in the first diffusion modes for larger values of $\epsilon$. In particular, we expect the most prominent coarse heterogeneity to be contained in the first independent diffusion mode.
For the chimera state depicted in Fig. 5.4(a), the first eigenvectors for $\epsilon = e^{-2.5} \approx 1.7 \cdot 10^{-3} D_{\text{max}}$ and $\epsilon = e^7 \approx 22.5 D_{\text{max}}$ are shown in Fig. 5.4(d). Note that for large $\epsilon$, $\phi^{(1)}$ possesses approximately the same value in the whole coherent region and in the whole incoherent region, respectively, with a continuous connection linking the two plateaus due the diffusion boundary between them. Due to the choice of $\epsilon$, it contains the dominant heterogeneity of the data at a very coarse degree, yielding a one-dimensional coarse-grained description of the time series. In short, $\phi^{(1)}$ seems to be clustered in the coherent and the incoherent region, respectively. This is in accordance with earlier studies, which have shown that the dynamics of the type-II chimera state are based on a modulated amplitude two-cluster state [95, 96].

Note that, like in the example of the pre-Bötzinger neurons, the Euclidean distances between points with similar $\phi^{(1)}$ are not necessarily small. In general, we argue that points being neighbors in $\phi^{(1)}$ means that they share the same behavior. In this case, they correspond to the same underlying cluster.

Furthermore, it is worth mentioning that for large $\epsilon$ diffusion maps no longer consider the physical space to be the dominant heterogeneity in the system. This becomes evident in Fig. 5.4(c), where the first two diffusion modes for $\epsilon = e^7 \approx 22.5 D_{\text{max}}$ are plotted. These modes no longer form a circle, but have multiple loops and thus no longer parametrize physical space.

When $\epsilon$ is very small, an alternate effect is seen, whereby each embedding vector $\phi^{(k)}$ becomes an indicator function on only one data point. Though such an outcome is pathological, there can be intermediate values of $\epsilon$ for which the indicator function interpretation is meaningful. Namely, another possible analysis outcome, is that diffusion maps might uncover a categorical variable rather than or in addition to the continuous variables seen here. This effect manifests in an embedding vector with a bimodal or multimodal distribution of values. Further, there might be a sharp peak in one mode, indicating only class membership, or alternately a slightly broader...
distribution, indicating a continuum of values within that class. The former is the case when all points in one cluster are very close to each other, but far from the rest of the data, while the latter occurs when the points in one cluster are still mutually distinguishable on the length scale implied by $\epsilon$, but effectively infinitely far from the rest of the data under the influence of the $\epsilon$-width Gaussian nonlinearity. This interpretation of the result of diffusion maps, whereby a variable might be deemed categorical or continuous depending on the length scale of analysis, is closely related to the method of spectral clustering [74, 73].

5.4 Reconstructing an attractor

In the previous examples, we compared time series to find variabilities which are intrinsic to the system, and used them as coordinates in which we regarded the data. But since we took whole time series as data points, the variabilities we could uncover were always perpendicular to the time direction and thus no variabilities along the time direction could be obtained.

In this section, we extend our approach by cutting each time series into time chunks shorter than the available time interval in our data, and using these as data points. In addition, we let these pieces overlap, which ensures that their distances to neighbors in space and time are comparable. This approach enables us to find variabilities also in the time direction, and will be illustrated in more detail on simulation data of the Kuramoto-Sivashinsky equation (KSE).

The KSE is a fourth-order partial differential equation that may be used to model spatiotemporal instabilities in a number of physical settings [12]. For the right parameter value [52], it exhibits a modulated (period-2) traveling wave, as shown in Fig. 5.5(a) for a 1d system with periodic boundary conditions. In this case, the
dynamics are defined by two oscillations: that of the underlying traveling wave itself and that of the modulation.

As a particular example, the dynamics shown in Fig. 5.5(a-b) are in the form of $N = 100$ time series of length $T = 500$. Our first $N$ time chunks are taken to consist of the first $l_{\text{chunk}} = 100$ time steps at each point in space, respectively. The next $N$ time chunks are similarly created by shifting the starting point to the $n$th time step of each time series and including the next $l_{\text{chunk}} = 100$ steps from there. This process is repeated until the last $N$ time chunks range from discrete time unit $T - l_{\text{chunk}} = 400$ to $T = 500$. Therefore, each time chunk overlaps with the previous one by $l_{\text{chunk}} - n$ time steps.

For example, when obtaining time chunks of length $l_{\text{chunk}} = 100$ from time series of length $T = 500$, a shift of $n = 8$ leads to the creation of $51 \cdot N$ overlapping time chunks. When performing diffusion maps on these time chunks the result, as in the previous sections, again depends on the value of the kernel scale $\epsilon$. For $\epsilon = e^{-0.5} \approx 0.10D_{\text{max}}$, a one-parameter dependence of the dynamics is detected, the diffusion coordinates of the time chunks forming a closed ring spanned by $\phi^{(1)}$ and $\phi^{(2)}$ (see Fig. 5.5(d)). As indicated by the smoothness of the coloring, the ring is parametrized by the spatial coordinates associated with the time chunks of any fixed starting point.

Decreasing the scale parameter, the ring in diffusion space partially unfolds into several rings, meaning that a third independent direction of the dynamics is gradually being detected. For $\epsilon = e^{-2.0} \approx 22.7 \cdot 10^{-3}D_{\text{max}}$, this is shown in Fig. 5.5(e). At $\epsilon = e^{-3.2} \approx 6.8 \cdot 10^{-3}D_{\text{max}}$ we get a full separation of the rings, as shown in Fig. 5.5(h). Together, these rings span a torus in diffusion space, one direction of which, the one along each of the rings, corresponds to the underlying traveling wave. The other direction, from one ring to the next, we argue as follows to correspond to the modulating oscillation: When considering the period of one modulating oscillation in Fig. 5.5(b), its length is found to be 56 time steps in the $T = 500$ discretization.
described above. For an increment in the creation of time chunks of \( n = 8 \), the time chunks starting at \( t_7 = 7n = 7 \cdot 8 = 56 \) will thus have the same phase with respect to the modulating oscillation as the time chunks starting at \( t_0 = 0 \). Moreover, the profile of the underlying unmodulated traveling wave stays the same for all times, and is only shifted in space. Together, this assures that each of the time chunks starting at \( t_0 = 0 \) is identical to one of the time chunks, at another point in space, starting at \( t_7 = 0 \) (barring numerical sampling inaccuracy). As the same argument applies to any collection of time chunks starting at a given \( t_i \), all these collections will thus occupy only seven distinct regions in diffusion space, and as each of them is represented by a ring, the result will be seven rings, arranged in a circle, exactly as in Fig. 5.5(h).

Analogously, choosing a different increment of \( n = 14 \) when creating the time chunks, leads to the diffusion modes shown in Fig. 5.5(i), where the torus is spanned by only four rings, just as predicted by our explanation. Moreover, the theory also holds for the choice of increments \( n = 28 \) and \( n = 4 \), leading to 2 and 14 rings spanning the torus in diffusion space, respectively (not shown here).

Note that the toroidal direction corresponds to the phase of the modulation, whereas the poloidal direction corresponds to the phase of the traveling wave. Thus, those two directions are the dominant variabilities in this dynamical system, compare Fig. 5.5(h) and (c).

## 5.5 Discussion

Comparing time series, or alternatively snapshots of spatial profiles, or alternatively local patches of spatiotemporal behavior, allows for rearranging and embedding them in a coordinate frame *intrinsically* to the data, and thus, to the dynamics. The examples above illuminate the different features of this approach. For spatiotemporal extensive chaos, such as the intermittency in the complex Ginzburg-Landau equation, one finds
coordinates which are correlated with the physical dimension(s) of the system. This leads to the conclusion that for this kind of extensive chaos, it is meaningful to sort the data along the actual spatial coordinates.

On the other hand, a more coarse grained description is often desirable in complex heterogeneous networks. In the heterogeneous model of pre-Bötzinger neurons discussed in §5.2, we are able to find variabilities which are one-to-one with the heterogeneous parameters $I_{\text{app}}^i$ and $\kappa_i$. As already discussed in earlier works [16], the dynamics in this space become effectively two dimensional. Note that, although one parameter, $I_{\text{app}}^i$, is intrinsic to the individual oscillators and the other heterogeneous parameter, the degree, is intrinsic to the network structure, both variabilities can be uncovered by the diffusion maps approach.

Furthermore, there are dynamical states in which a very coarse, and possibly one-dimensional, description of the system is desirable. An example are chimera states, states of coexisting coherence and incoherence. There, one often relies on one-dimensional order parameters to describe the order and disorder of these states. Here, we achieve this very coarse description by choosing the kernel scale large. This enables us to find order parameters which are homeomorphic to previously calculated order parameters of the chimera state in an ensemble of nonlocally coupled phase oscillators. Even for chimera states with strong amplitude effects, it is possible to extract meaningful order parameters, which, as we argue, will facilitate the further investigation and characterization of these intricate states.

Basing the analysis on time or space chunks, or space-time patches, we obtain information which is local both in space and time. Provided these strings or patches are large enough, they contain information about their location on the underlying attractor of the dynamics. In our last example, we exploit this property, and show that it is possible to find the attractor of modulated traveling waves in the Kuramoto-Sivashinsky equation. Using overlapping time chunks, overlapping spatial strings or
space-time patches, we find diffusion modes that span a space in which the dynamics are projected onto a torus.

These examples lead to the general conclusion that we can construct a suitable space from the dynamic behavior itself, a space in which the data is ordered according to its intrinsic variability.

Yet the observer is still there - in the choice of the quantities observed, and in the scale of the diffusion maps kernel (the scale at which the observer is sensitive to -can perceive- quantitative changes in the observations, as discussed above in the study of the chimera states). The exact value of the kernel scale $\epsilon$ depends on the context: if the scientist is interested in the details of the system, a more refined representation of his/her data is desired, and thus a small kernel scale. In contrast, a broader kernel may be desirable if one is interested in a very low-dimensional representation of the present measurements.

Since we have mainly focused here on processing simulation data, our observables were the full system states; remarkably, suitable embeddings can also be obtained even if the full state is not accessible: if only partial (even very partial!) measurements of the system state, or of (invertible) functions of the system state, can be observed. When the dynamic behavior lies on a smooth low-dimensional (assume $m$-dimensional) manifold, it has long been established that this manifold can be smoothly embedded in a $2m$-dimensional space (this is the celebrated Whitney embedding theorem [117, 94]). In the context of dynamical systems, F. Takens and N. Packard et al. established that a $d$-dimensional attractor can be reconstructed in a $k > 2d$ dimensional Euclidean space [105, 82]. So a sequence of observations of the system state, even observations of a single observable, can –through the use of time-delays– exploit the measurement history to reconstruct the attractor. Perhaps even more celebrated are the Nash embeddings theorems, which allow for the reconstruction of isometric embeddings of the low-dimensional manifold on which the behavior evolves from “a few” observations
of the system state (or of smooth functions of the system state, or of “delays” of one or more observations) [75, 76].

Embeddings of different (sufficiently rich) sets of observations of the same system will, in general, be different, even if comparably informative. As a case in point, consider the observation of sunrise/sunset in German cities not through the Boolean vectors of the daylight times over the hours of a particular day of the year, but rather through vectors containing the sunrise and sunset times of an entire year, as shown in Fig. 5.6(a). Here again, diffusion maps yield a similar (one could argue, homeomorphic) arrangement of the cities, as depicted in Fig. 5.6(c).

Clearly the two types of measurements are related to the same phenomenon, and thus to each other; and clearly both reconstructed city arrangements are reminiscent of the true map of Germany. Yet this begs the question: is there a systematic way of processing the data from different observations of the same problem so as to result in the same embedding? In other words, can one perform gauge invariant data mining - processing of the data in a way that is insensitive to the observation function? More concisely, is there a data mining procedure invariant to diffeomorphisms?

In the context of data mining, the (positive) answer was given by Singer and Coifman in 2008 [100]; they proposed a diffusion kernel based on a so-called Mahalanobis-like distance, leading to what they termed Nonlinear Independent Components (or Nonlinear Independent Variables, “NIVs” [29, 30]). These global embedding variables are, to high order, insensitive to the observation function.
Figure 5.5: (a) Temporal evolution of $u$ in the 1d Kuramoto-Sivashinsky equation with periodic boundaries, showing modulated traveling waves. (b) The dynamics in (a) in a co-rotating frame. The color corresponds to $u$. (c) Torus spanned by the phase with respect to traveling wave, $\theta$, and modulation, $\zeta$, respectively. Going along space (red line) only changes the former, whereas moving in time (green line, dashed blue line) changes both phases. (d) First three independent diffusion modes obtained from overlapping time chunks created with a time shift of $n = 8$ and for $\epsilon = e^{-0.5} \approx 0.10 D_{\text{max}}$, colored by spatial position. (e-h) Equivalents to (d) for $\epsilon = e^{-2.0} \approx 0.0227 D_{\text{max}}$, $\epsilon = e^{-2.62} \approx 0.0122 D_{\text{max}}$, $\epsilon = e^{-2.627} \approx 0.0121 D_{\text{max}}$ and $\epsilon = e^{-3.2} \approx 0.0068 D_{\text{max}}$, respectively. (i) First three independent diffusion modes obtained for $n = 14$ and $\epsilon = e^{-2.3}$, colored by spatial position.
Figure 5.6: 5.6(a) Sunrise and sunset times in GMT of different German cities [3].
5.6(c) The first two independent diffusion modes, $\phi^{(1)}$ and $-\phi^{(2)}$ for $\epsilon = e^{15}$ of the data shown in 5.6(a), colored with their longitudinal geographic position. Note that by comparing with the actual locations in 5.6(b), it becomes obvious that $\phi^{(1)}$ and $\phi^{(2)}$ span a space in which the mutual arrangements of the cities reflect their actual positions. Additionally, due to the correspondence between the coloring and $\phi^{(1)}$, one can visually verify that $\phi^{(1)}$ parameterizes the west-east direction.
Chapter 6

Circadian rhythms

In this chapter, we examine the application of some of the methods described earlier in this dissertation to a complex biological model for circadian rhythms in the central nervous system. Circadian rhythms are the intrinsic pattern of daily oscillation experienced by a wide array of organisms, from mammals to plants. In mammals and elsewhere, they are governed by a core clock, an interconnected population of a few hundred or thousand neurons in the central nervous system, each which expresses a set of core clock genes in a rhythmic pattern each day. In humans, this population of neurons is located in the suprachiasmatic nucleus (SCN) of the hypothalamus. Even in culture, isolated from external light, drug, or neurotransmitter signals, many of these neurons will continue to oscillate with a period of around one day—circa-dian.

Disorders of circadian rhythms are the topic of several recognized syndromes in humans, including both early- and late-awakenings, and non-24-hour sleep/wake disorder. Additionally, some proposed shift work patterns are designed to require a non-24-hour sleep/wake cycle. Motivated by applications in both of these situations,

The biophysical model described in this chapter was developed jointly with the lab of Michael Henson of the University of Massachusetts, Amherst.
we seek realistic and efficient modeling capabilities for mammalian circadian rhythms and possible therapeutic interventions thereto.

The model used here builds on previous work, with new developments and appropriate background described in §6.1. We first examine the basic set of single-cell differential equations on which the model is built, then move on to describe and cite additional work that was done to model coupling between cells using two different neurotransmitters. We finish the section by considering more recent work to incorporate the influence of both external signals through both light and drug dosing, both of which are candidates for developing control strategies.

Moving on from the specifics of the model developed, we view the results of applying some numerical methods for nonlinear dynamics to the SCN model. We consider general methods for finding forced and unforced limit cycles through the discovery of fixed points of stroboscopic and Poincaré maps, and show the application of the former to the light-forced SCN model. We extend this analysis by using pseudo-arclength continuation to find the locus of stroboscopic fixed points with varying forcing period. We then repeat this continuation for several forcing strengths, producing an approximate resonance horn portrait for the SCN model.

Drawing on public data detailing experimentally recorded bioluminescence time-series for a population of circadian neurons [4], we repeat the equal space analysis of chapter 5, and show that the population of neurons, viewed by these timeseries, well-described as a two-dimensional population, with coordinates that appear to reduce to the phase shift and amplitude of the single-cell traces.

Finally, we observe the evolution of a polynomial chaos expansion (PCE) description for a simulation of a population of SCN neurons, as described in detail in chapter 3. It can be seen that the PCE fit describes the population behavior well at moderate polynomial order, effecting a reduced dimension description of the system state.
6.1 Background biology

ODEs and previous work

The base model [68, 69], developed by Leloup & Goldbeter in 2003-2004, simulated a single cell with 16 ODEs describing the evolution of the mRNA, proteins, and protein-protein complexes for the genes Per, Cry, and Bmal1.

In 2007, To et al. [110] added one ODE per cell for cyclic AMP response element binding factor (CREB), accounting for vasoactive intestinal peptide (VIP) signaling. VIP concentrations were made to be a direct function of Per mRNA concentration (MP). This allowed for the simulation of a network of neurons. The network used was all-to-all, with edge weights reciprocallly weighted by Euclidean distance in a 2D plane. Vasalou et al. in 2009 [114] used the same $17n_{\text{nodes}}$ equations, and investigated the effect of using a small-world topology.

Vasalou and Henson in 2010 [112] then used firing rate to link hour-timescale gene expression dynamics with millisecond-timescale neurotransmitter output. This required explicit modeling of the dynamics of calcium, leading to two new ODEs. They also introduced an ODE for VIP, rather than making it a direct function of Per mRNA. In a 2011 paper [113], Vasalou & Henson differentiated between the core and shell regions, providing a detailed topological model of the whole rat SCN. Finally (also in 2011), Vasalou, Herzog, & Henson [115] added another ODE for $\gamma$-aminobutyric acid (GABA) signaling (earlier just an algebraic function of the other state variables), which should lead to a total of $21n_{\text{nodes}}$ ODEs.

Below, we describe the model which has since been built on the basis of this collective body of work.
Signaling

In [115], ODEs are given for GABA and VIP neurotransmitter levels released by each neuron, as a function of local intracellular gene expression levels. The corresponding dynamic variables, $GABA_i$ and $VIP_i$, are therefore defined with indices $i$ across cells. The influences of these on neighboring cells are given by summation, restricted by two adjacency matrices $W_{VIP}$ and $W_{GABA}$. We calculate the normalizing row sums of these matrices as

$$\epsilon_{M,i} = \begin{cases} 0 & \text{if } \sum_j M_{i,j} = 0, \\ 1/\sum_j M_{i,j} & \text{otherwise.} \end{cases}$$

for $M = W_{GABA}, W_{VIP}$.

For GABA, we let $GABA_{f,i}$ be the functional (non-background) GABA concentration for which we have an ODE at each neuron, and $GABA_i = GABA_{f,i} + p_{GABAr}$ be the total observed concentration released by cell $i$, where the parameter $p_{GABAr}$ is the released background level. We then let $GABA_{s,i} = p_{GABAs} + \epsilon_{G;i} \sum_j GABA_j A_{G;i,j}$ be the total GABA saturation observed by neuron $i$, where, $p_{GABAs}$ is the saturation background level of GABA. This GABA saturation is then used in the the expression for the pseudoequilibrium concentration of chlorine inside the cell, as defined in [115] (SI, Eqn. 12), and derived from the open channel probability in Eqn. 4 of [92]

$$P_{open} = \frac{1}{1 + L_0 \left(\frac{1+[GABA]/K_G}{1+[GABA]/cK_G}\right)^2 \left(\frac{1+[BZ]/K_{BZ}}{1+[BZ]/dK_{BZ}}\right)}, \quad (6.1)$$

which includes also a dependence on local gated benzodiazepine saturation [BZ] (discussed further in §6.1).

For VIP, we let $VIP_{s,i} = \epsilon_{A;i} \sum_j VIP_j A_{V;i,j}$ be the total synaptic VIP concentration observed by neuron $i$. We then combine this with a second VIP saturation variable derived from the the light input level.
Specifically (dropping the index \(i\) in the remainder of this subsection), we define the intermediate quantities \(s_{\text{VIP}} = \rho_{\text{VIP}}/(K_D + \rho_{\text{VIP}})\) (from [110], with \(s_{\text{VIP}} \in [0, 1]\) and \(K_D = 0.06\)), and \(w = \expit(s_{\text{VIP}} + SG - 1)/4\), where the sigmoid \(\expit(x) \overset{\text{def}}{=} 1/(1 + e^{-x})\) transitions sharply from 0 to 1 as its argument passes through 0. The variable \(SG\), which goes low in daylight and high at night, is described below. However, the effect of light input on VIP receptor saturation is additionally modeled through the quantity

\[s_{\hat{\text{VIP}}} = s_{\text{VIP}}(1 - L) + L(wL + (1 - w)(s_{\text{VIP}} + SG)).\]

\(s_{\hat{\text{VIP}}} \in [0, 1]\) represents the combined contribution to VIP saturation from both VIP and light. This quantity enters into equation (30) in [112] (a modification of (5b) of [110]), which calculates a rate of protein kinase activity \(v_k\). This kinase rate is used in (5a) of [110], the ODE for \(CB^*\) (phosphorylated CREB)

\[v_k = V_{MK} \frac{C_{\text{tot}}}{k_{MK} + C_{\text{tot}}} + V_b \frac{s_{\hat{\text{VIP}}}}{k_b + s_{\text{VIP}}},\]  

where \(C_{\text{tot}} = x_a C_{\text{Cytoplasm}}^2 + w_{ow}\), \(x_a = 1\) is a possible multiplicative scalar, and \(w_{ow} = 0\) is a possible background concentration.

**Light input**

Light is an effective mechanism for producing phase advances or delays to account for rotating shift work. Light sensitive neurons in the SCN secrete glutamate and VIP, which has numerous excitatory effects on receptor neurons [113].

In the original model of Leloup & Goldbeter [68, 69], the effect of light, which was assumed to be a Boolean forcing, was modeled as a multiplicative increase in \(v_{sP_{\text{max}}}\), the maximum rate of \(Per\) expression. When VIP signaling was added in later work [110, 114], light was assumed to act through the same cascade as VIP signaling: receptor activation led to changes in intracellular calcium concentration, leading to
activation of Per transcription through the protein CREB. Light was still assumed to be Boolean, and so VIP release rate was modeled with a function of the state variable MP during darkness but with a constant high value during light. The same mechanism was used when GABA was added later [113].

Although light was previously modeled as a Boolean switch (it is either on or off), experiments [23] show that the effect of light on phase shifting is reduced during prolonged continuous exposure, motivating the modeling of a “saturation gate” (a 22nd ODE for each cell) [107], denoted by SG. The effect of light is strongest during the first two hours of exposure, followed by an exponential decay to its minimum value of 22 percent of the initial effect.

SG ("saturation gate") is a dynamic variable with exponential (linear) decay towards a set point, i.e. \( \frac{dSG}{dt} = r(SG - x_0) \). Both the decay rate \( r \) and set point \( x_0 \) are determined by treating light level \( L \in [0, 1] \) as an indicator function, i.e., \( r = Lr_l + (1 - L)r_d \) and \( x_0 = Lx_{0,l} + (1 - L)x_{0,d} \). This formulation works whether \( L \) varies continuously, or switches abruptly from 0 to 1.

We avoid discontinuities such as the hard transition from \( L = 0 \) to \( L \neq 0 \) at dawn by smoothly transitioning between such states. That is, light level \( L \) is allowed to be 0 during most of the night, and 1 during most of the day, with a smooth sigmoid transition (shaped like expit(\( t \))) during the twilights. (Thus, a suitable forcing function might be expit(10 sin(2\pi t/24)), though true light-level timeseries, parameterized by date and geographic location, could also be used.)

Additional modulation of chloride potential by daily circadian rhythm

In addition to the effect of light input on the gene oscillator through the saturation gate, VIP saturation via \( \hat{\beta} \), and the resulting effect on phosphorylated CREB concentration, experiments have observed daily chloride oscillations [115, 113], and the parameters in
our equation for intracellular pseudoequilibrium chloride concentration are designed empirically to bring those oscillations in the model in line with their observed phase and range. That is, $C_{\text{in}}$, for each cell, is written as a function of Per mRNA ($MP$) in addition to GABA saturation.

$$C_{\text{in}} = Cl_o + \frac{v_{\text{Cl}_1} MP}{K_{\text{Cl}_1} + MP} + \left(1 - \frac{1}{1 + L_{0\text{Cl}_1} (1 + G_s/c_{\text{Cl}})^2}\right) v_{\text{Cl}_2} \quad (6.3)$$

Here, $G_s = GABA_s \cdot 10^5/K_G$. The specific form of the $MP$ influence $ux/(K + x)$ is a Hill-type equation which resembles the kinetics of receptor saturation. We are not implying a direct mechanistic relationship between $MP$ and chlorine—it may be that there is an intermediate in phase with $MP$ that is responsible for oscillations in $C_{\text{in}}$.

### Drug Dosing

Though not used for the simulations shown in this chapter, the model as implemented includes a mechanism for incorporating the effects of dosing with benzodiazepine (BZ) drugs, such as midazolam or triazolam. We incorporate an equation describing the pharmacodynamics of midazolam (in terms of a modified probability of opening for GABA-gated chloride channels) [92]. This includes nominal parameter values experimentally determined by Rüsh and Forman. Since other benzodiazepines (ramelteon and triazolam) have been found capable of producing an wheel-running phase shift in a mouse genetic model of early morning awakenings [118], this drug target is viewed as a promising candidate for perturbing circadian rhythms.

While these studies [92, 118] provide useful calibration information for the pharmacodynamic effect of BZ on the circadian clock, we have used information from other sources to tune the pharmacokinetics by which the drug is cleared from the body. Our model approximates the PK for oral ingestion [46] whereby saturation-gated BZ concentration increases at constant rate for approximately half an hour from the time
of ingestion, holds constant while dosing is still active, and then decays exponentially back to zero when dosing is removed. In accordance with these results, BZ is 80 percent cleared after four hours. Oral ingestion of BZ is assumed most practical from the perspective of rotating shift workers.

6.2 Finding limit cycles

As this dynamic system is a (population of) autonomous oscillators, one useful computational task to perform is the acquisition of the limit cycle, or one point on it. On the other hand, circadian oscillations are one of the oldest studied examples of a forced oscillators—though they have a natural period (of the lumped population of cells), they are externally driven by an independent oscillator with no returning feedback (the sun).

In experiment and sometimes in simulation, these invariant features (limit cycles) are often found by direct integration—the ODEs are integrated or the experiment run for a long time, until it appears that the system has settled into a rhythm. At apparent equilibrium, the stable state is characterized by various statistics, like the time between peaks, or the power spectrum of the stable wave. While this method works well for experiments, it naturally can only discover stable invariant features. For simulations, this method can be significantly slower than more direct methods described below.

We provide forcing by allowing light level $L(t)$ from §6.1 to vary continuously over the “day”. The light waveform is square-wave like, but transitions are instead replaced by smooth sigmoid shapes. $L(t)$ varies from 0 at night to some maximum value ($\in (0,1]$) during the day, which is our amplitude parameter $A$. The period of the near-square wave is our period parameter $\tau_f$. 

96
Poincaré and stroboscopic maps

For autonomous oscillations, finding a limit cycle is often reduced to finding a single point on the limit cycle, and particularly a fixed point of a Poincaré map. Such a map observes the flow of the system after time $\tau$, but only on its projection to a codimension-1 subspace of the full $n_{\text{fine}}$-dimensional space. Values of $\tau$ and the remaining $n_{\text{fine}} - 1$ state variables are then sought which are a fixed point of this map, the final missing degree of freedom of which is provided by measuring the actual time required to integrate from initial projection onto the subspace to a specific stopping condition. [42]

In practice, the subspace is often formed by taking a “Poincaré cut”, in which one of the variables (say, index $k$) has its value fixed to a known typical value for the oscillation sought. For an arbitrary length-$(n_{\text{fine}} - 1)$ initial vector $\hat{x}(0) = \{x_j(0)\}_{j \neq k}$ and period guess $\tau$, the starting vector is expanded with the cut value $x_k$ to be a full state $x(0)$. The simulation is then run forward from this state until time $\tau_r$, when the cut is reached again in the same direction as we left it (often this will involve crossing the cut once roughly halfway through our simulation). To gain precision, further simulations can be performed recursively starting and stopping on the last two timesteps until a final state is within some tolerance of the stopping condition, and/or a linear interpolation can be used.

We then reduce the final state $x(\tau_r)$ to $\hat{x}(\tau_r) = \{x_j(\tau_r)\}_{j \neq k}$, and return this reduced state with the true required time $\tau_r$. The caller can then subtract the final reduced state to the initial/provided state to get a residual vector, and subtract the final time $\tau_r$ from the predicted time $\tau$. Solvers such as Newton’s method (or Newton-Krylov GMRES for large $n_{\text{fine}}$) can now be used here to find the true fixed point and true period.

This procedure is necessary for finding autonomous limit cycles, whether stable or unstable. However, when a non-autonomous forcing signal is provided, the situation
is simpler, as we know a priori what the period of the oscillation will be (the forcing period $\tau_f$). In this case, we simply seek a point $x(t_0)$ equal to its image in the time-$\tau_f$ flow map $\Phi_{\tau_f,F}$, defined as in Eqn. (2.1). Since the forcing makes time unique, there is no ambiguity in the direction locally collinear with the limit cycle–for a given $t_0$, there is only one solution. In effect, we are solving a Poincaré map where the cut variable is the phase $t/\tau_f$ of the forcing.

In the forced case, where we solve a stroboscopic map, it is possible additionally to solve the so-called variational equations, ODEs for the evolution of a matrix $V(t)$ over the course of the oscillation [106].

$$V(t) \overset{\text{def}}{=} \frac{\partial x(t)}{\partial x(0)}$$

$$\frac{d}{dt} V = \left[ \frac{\partial f}{\partial x} \right] \cdot V$$

$$V(0) = \left. \frac{\partial x(t)}{\partial x(0)} \right|_{t=0}$$

$$= I \quad (6.4)$$

A solution of these equations, evaluated at $t = \tau_f$, is called the \textit{monodromy matrix}. This matrix is somewhat analogous to the Jacobian for fixed points of dynamical systems, in that its eigenvalues can be used to determine the stability of the invariant feature in question (fixed points of the stroboscopic map, which are observations of the underlying limit cycle). Namely, if all the eigenvalues of $V(\tau_f)$ lie within the unit circle in the complex plane, the stroboscopic fixed point on the limit cycle is deemed stable.

Such a solution for the light-forced circadian oscillator is shown in Fig. 6.1. The oscillation covers a reasonable domain (the plotted quantities are in units of concentration, and so should be positive, and the found solution at this moderate period of $\sim 23$ hours is stable. In the next subsection, we will examine the dependence of this portrait on the strength and period of the forcing signal, $A$. 

98
Forcing with strength $A = 0.85$ and period $\tau_f = 22.9 \text{ [h]}$. Largest-abs Floquet multiplier is $|0.88| < 1$ (stable).

Figure 6.1: Limit cycle for a circadian oscillator system of two neurons forced by a sigmoidally varying light signal

**Isolas and resonance horns**

With a stroboscopic map to solve, we can look at the dependence of the solution on one of the two parameters of the forcing. To examine the dependence of the portrait of Fig. 6.1 on forcing period $\tau_f$, we perform a pseudo-arclength continuation. This produces a figure like Fig. 6.2. Individual points along this ring of solutions correspond to the fixed points found on limit cycles like that in Fig. 6.1. Both stable and unstable solutions exist, though these might be opposing points on the same limit cycle. For small and large values of $\tau_f$, these points collide, and the solution disappears. We refer to this circular branch of solutions as an “isola”.

Pseudo-arclength continuation is a predictor-corrector method, which takes place in two parts for each step along the branch. In the predictor part of the step, an approximation to the tangent vector at the previous solution is used to predict the location of the next branch point a specified distance $\Delta s$ forward in arclength along the curve. This approximation can use finite differences, or, for a any but the first step, simply the chord between the last two branch points. In the corrector part of the
Figure 6.2: Isola of stable and unstable stroboscopic map solutions, for several forcing periods at a fixed forcing amplitude
step, this guess is supplied to a nonlinear solver whose objective is constructed such that both (1) the solved-for point is in fact a fixed point of the stroboscopic map, and (2) the projection of the step onto the tangent vector is of length $\Delta s$. For a system defined in $n$ variables, the task at each branch point is to solve for $n + 1$ quantities, including the continuation variable (here, $\tau_f$). This extra degree of freedom makes the problem of the corrector part well-posed, even with its extra requirement of length-$\Delta s$ projection.

In the regime of $\tau_f$ and $A$ values for which isolas like Fig. 6.2(a) exist, we can additionally look at the dependence of this portrait on forcing strength $A$. While a further continuation scheme is possible in principle (appending a further condition that one eigenvalue of the monodromy matrix, or one complex conjugate pair, be on the unit circle) here, we simply repeat the computation of Fig. 6.2 for several different values of $A$. With such a scheme, we compose Fig. 6.3. We can see that this method composes an approximate resonance horn, or “Arnol’d tongue” [106]. In this diagram, points on the interior of this structure allow for entrained limit cycles, while points outside allow only quasiperiodic solutions, since the forcing period and the natural are too disparate for the weak forcing strength to overcome.

### 6.3 Equal space

We now reproduce in part the “equal space” analysis of chapter 5 on experimental data recorded via producing cells that luminesced in tandem with expression of the PER gene [4]. In these recordings, tetrodotoxin (TTX) was used to interrupt signaling between cells for a period of several days in the middle of the experiment, while allowing endogenous oscillation to continue. After this, TTX was washed out, allowing the cells to resynchronize. This data is depicted in Fig. 6.4(a).
Figure 6.3: Approximate Arnol’d tongue/resonance horn constructed from several isolas at different forcing amplitudes
Figure 6.4: 6.4(a): Data from [4], divided into “desynchronized” (blue) and “resynchronized” (red) portions. The authors of [4] recorded PER::LUC luminescence during several days, and used tetrodotoxin (“TTX”) to block interneuron communication during the period marked in blue. 6.4(b): An analogous simulation on simple Kuramoto oscillators, as described in chapter 3. After neurons synchronize, coupling is briefly removed, then restored. Modulo $2\pi$, the arrangement resumes its previous configuration.

We can produce analogous data for a heterogeneous Kuramoto simulation (as described in chapter 3), with or without coupling, as seen in Fig. 6.4(b). When wrapped into a $\pm \pi$ viewing window, it is clear that the Kuramoto oscillators quickly desynchronize and resynchronize when coupling is removed and re-imposed.

Using cell trajectories only from the re-synchronized portion of the dataset shown in Fig. 6.4(a), we perform an equal space analysis of the kind described in detail in chapter 5. We perform the analysis both on the full per-cell trajectories in this window (results shown in Fig. 6.5), as well as on a subset of a larger database of short snippets from this window (results shown in Fig. 6.6).

In both cases, we plot the uncovered eigenvectors as $\phi^{(j)}$ vs $\phi^{(k)}$ in a grid over $j$ and $k$ values. The plots are colored by the residual norm for their locally-linear regression reconstruction in terms of lower-indexed eigenvectors, as described in detail in [27]. This residual quantifies how novel each subsequent eigenvalue is—lower-residual vectors are likely to be harmonics of earlier higher-residual vectors, and hence carry no new information. As such, we can see that both equal-space deconstructions are...
Figure 6.5: Data from the red-boxed part of Fig. 6.4(a), analyzed with the “equal space” method of chapter 5. The method uncovers phase and amplitude variation in the signals.

Figure 6.6: Data from the red-boxed part of Fig. 6.4(a), chopped up into short segments and then analyzed with the “equal space” method of chapter 5. The method appears to uncover one increasing/decreasing parameterization and one concave-up/concave-down parameterization.
strongly two-dimensional. It is clearly visible in Fig. 6.5(b) that these two dimensions correspond to the phase-offset and amplitude of each cell’s trajectory.

### 6.4 PCE expansion

Continuing in the spirit of replicating techniques from earlier chapters on the circadian dynamical system, we now turn to observing a polynomial chaos expansion (PCE) coarsening on circadian dynamics. A simulation of the system described in §6.1 is used to produce a $23n_{\text{nodes}}$-dimensional timeseries. From this, we extract the timeseries for only the first of the 23 variables, across all $n_{\text{nodes}}$ cells, producing Fig. 6.7(a). As in Fig. 4.7, the individual per-cell trajectories show some variety, that unsurprisingly can be explained by two parameters which we deliberately have made heterogeneous. Parameters are sampled from a truncated Gaussian distribution, with a mean set to the model’s nominal value for that parameter, standard deviation of 0.1, and truncation bounds set to 0 and the nominal value times 10. These sampled distributions are depicted in Fig. 6.7(b).
Figure 6.7: Simulation data and PCE fit. Fig. 6.7(a) shows the last few cycles of the evolution of the per-cell variable under analysis obtained from simulation under the influence of the heterogeneities depicted in Fig. 6.7(b). Fig. 6.7(c) shows a PCE fit at a particular time marked by a vertical red line in Fig. 6.7(a). Fig. 6.7(d) shows these fits at several other times during approximately the last cycle of the simulation.
As before, we see that the detailed state of the population of neurons can be approximated by a smooth polynomial surface. We can see the evolution of these fitting coefficients for the chosen per-cell variable in Fig. 6.8(a). How well the fit surface captures the states of all the neurons is examined in Fig. 6.8(b)—already, the absolute maximum fitting error is small relative to the typical values of the variable at a polynomial order of 3.

Figure 6.8: PCE fit over time for SCN simulation data. Fig. 6.8(a) shows the fitting coefficients themselves over time for one chosen polynomial order, while Fig. 6.8(b) shows the error of the approximation for several different possible polynomial orders.

While this analysis can be applied to each of the 23 per-cell variables, here we focus on the Per mRNA. Repeating this analysis for the other per-cell variables would produce a number of coarse variables equal to the number listed in the legend of Fig. 6.8(b) multiplied by 23—still less than the full 23\(n_{\text{nodes}}\). Likewise, we might use a more general set of heterogeneous parameters, chosen from the enormous set of parameters in the model equations given in [69, 110, 112] and §6.1. However, we focus on two kinetic constants, \(k_{sB}\) and \(v_{sB}\). An alternative approach would be to allow heterogeneity in a large set of parameters, but employ the equal space method of chapter 5 to obtain a reduced parameter space.
6.5 Conclusion

In this chapter, we described a computational model for simulating the underlying neural dynamics of circadian rhythms. We began by summarizing the biology of the model, in both previous work and new developments, and mentioned possible targets for therapeutic intervention that are included in the simulation. We continued by examining the results of several numerical explorations of the nonlinear dynamics of this simulation. Shifting gears, we saw that the equal space method from chapter 5 can be applied to experimental recordings of this system, with informative results. Returning to the simulation, we saw that the polynomial chaos expansion of chapter 3 is also applicable here, and capable of summarizing the system state to high accuracy while reducing the dimension.

At the time of this writing, development of both the model and its applications remain underway, with active collaboration producing enhancements to the model and new applications for the numerical methods. In future work, we plan to use the drug and light dosing capabilities of this model to generate open-loop control schemes for the treatment of both sleep disorders such as early awakenings, and to assist with entrainment to non-24-hour shift work schedules.
Chapter 7

Conclusion

In this dissertation, we examine methods for coarse graining simulations of large heterogeneous populations. All the simulations considered comprise a population of individual agents, or units, whose states evolve over time under the influence of their network neighbors and sometimes an independent external forcing signal. A key detail is that the units may differ in both their dynamic state and their non-dynamic heterogeneous characteristics. In chapter 2, we introduced several methods used throughout the dissertation, as well as corresponding notation. These methods included coarse graining, polynomial chaos expansion (PCE), and diffusion maps.

Subsequently, we examined the application of these methods to several systems of increasing complexity.

We began in chapter 3 with a simple phase oscillator system, in which system state for each unit was simply a phase angle $\theta$. For this simple system, we looked at the application of numerical methods for analyzing dynamical systems, including the computation of fixed points (steady states) and the continuation of branches of these steady states as a global system parameter $K$ is varied.

In chapter 4, we returned to the math of PCE by exploiting the orthogonality of the basis to compute expansion coefficients directly by inner products integrals. Antic-
ipating high-dimensional heterogeneity, we repurposed analysis of variance (ANOVA) to produce a numerical scheme for computing these integrals that relies only on the integration of lower-dimensional functions. By way of application, we now considered a Hodgkin-Huxley-like simulation, with two dynamic variables per unit.

In chapter 5, we continued our examination of the Hodgkin-Huxley simulation, but now we pretend to have forgotten the nature of the heterogeneities differentiating the individual units. Instead, we record a simulation, and then employ diffusion maps to uncover the $d_u$-dimensional submanifold of trajectory-space on which the individual traces of the per-unit dynamic variables lie. We find that the discovered data-driven coordinates span the same space as the actual heterogeneities, and can be used to generate a reasonable smooth PCE coarse-graining.

Finally, in chapter 6, we bring these methods to bear on a much more complicated model for circadian rhythms. This model has up to 23 variables per unit, describing quantities such as mRNA, protein, and protein-protein complex concentrations in both nucleus and cytoplasm. These quantities interact with each other to produce an unforced rhythm with a period close to 24 hours, with individual simulated cells communicating through both vasoactive intestinal peptide and $\gamma$-aminobutyric acid signals. In addition to showing that the dynamics of the coupled network admit a PCE coarsening scheme, and solving a Poincaré map to find the unforced limit cycle of the whole system, we apply an external forcing rhythm through simulated light exposure, and examine the system’s response to forcing rhythms of various periods and strengths (amplitudes). We see that the entrainable region has the expected resonance-horn shape, with the system being unable to entrain to forcing signals that are too weak or whose period differs too much from the unforced period.

In summary, the contribution of this dissertation is to show the versatility of polynomial chaos as a coarse-graining technique for the coarse-graining of simulations of heterogeneous populations. We enhance several components of the PCE method
by the repurposing of related techniques: We generate basis functions from scratch for unknown or sampled distributions, rather than relying on known distributions with tabulated polynomials. We describe this moments-based method in detail for the general multidimensional case in §2.2, and apply it in the independent multidimensional case in chapter 3. Alternately, while in chapter 3 we rely on knowledge of the relevant population heterogeneities to construct our PCE coarsening, in chapter 5, we describe a novel data-mining based approach to obtaining sufficient heterogeneities directly from recordings of the population dynamics. By reducing the model-specific information required of the practitioner, these two enhancements bringing the method closer to being a black-box treatment for the coarse-graining of existing heterogeneous-population simulations. Simultaneously, the work described in chapter 4 represents a performance enhancement that increases the range of applicability of the PCE method. Namely, the poor scaling of the explicit PCE expansion with the number of heterogeneous dimensions $d_u$ is mitigated by expressing the necessary inner products (or expectations) not as high-dimensional integrals, but as the sum of a sequence of lower-dimensional integrals. By exploring these enhancements and generalizations of the PCE method, as transplanted from its original use in uncertainty quantification to this new application in “heterogeneity quantification”, we hope to encourage practitioners to include PCE in their toolbox for simulating large heterogeneous populations.
Bibliography


[91] Jonathan Rubin and David Terman. Synchronized activity and loss of
synchrony among heterogeneous conditional oscillators. SIAM Journal on Applied

[92] Dirk Rüsch and Stuart A Forman. Classic benzodiazepines modulate the open-
close equilibrium in alpha1beta2gamma2L gamma-aminobutyric acid type A


nr. 91017.

[95] Lennart Schmidt and Katharina Krischer. Chimeras in globally coupled oscil-
latory systems: From ensembles of oscillators to spatially continuous media.

[96] Lennart Schmidt and Katharina Krischer. Clustering as a prerequisite for
2015.

[97] B. I. Shraiman, A. Pumir, W. van Saarloos, P. C. Hohenberg, H. Chaté, and
M. Holen. Spatiotemporal chaos in the one-dimensional complex Ginzburg-


[99] Amit Singer and Ronald R. Coifman. Detecting the slow manifold by anisotropic

[100] Amit Singer and Ronald R. Coifman. Non-linear independent component
analysis with diffusion maps. Applied and Computational Harmonic Analysis,

Detecting intrinsic slow variables in stochastic dynamical systems by anisotropic
diffusion maps. Proceedings of the National Academy of Sciences, 106(38):16090–
16095, 2009.

[102] Per Sebastian Skardal, Edward Ott, and Juan G Restrepo. Cluster synchrony in
systems of coupled phase oscillators with higher-order coupling. Physical Review

and their Monte Carlo estimates. Mathematics and Computers in Simulation,


